

BSc FINAL YEAR THESIS PROJECTS

Proposed by Simon Winberg

Radar Remote Sensing Group

2015

Topics on the web at:

<http://www.rmsg.ee.uct.ac.za/members/swinberg/Proposals>



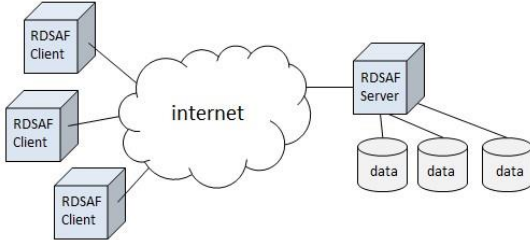
(The web version includes a convenient topic listing and hyperlinks.)

Images on the top right indicate the theme

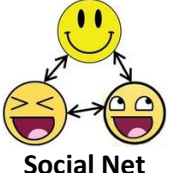
List of Topics

BSC FINAL YEAR THESIS PROJECTS	1
CATEGORY: SOFTWARE DEVELOPMENT & BIG DATA	3
SW-1 (R)	3
DSSAF WEB INTERFACE AND STREAMING CLIENT	3
SW-2 (R)	5
TUTOR ORGANIZATION AND PLANNING APP (TOPA)	5
SW-3	7
PTNS – PORTABLE TRADING NETWORK SYSTEM	7
CATEGORY: EMBEDDED SYSTEMS, FPGA, SIGNAL PROCESSING	8
SW-4	8
FERS SIMULATOR GUI UPGRADE	8
SW-5	9
ETHERNET INTERCONNECT ABSTRACTION BETWEEN RASPBERRY PI AND FPGA COPROCESSOR (<i>CODENAME EIARF</i>)	9
SW-6	10
FPGA DATA BLASTER KIT (<i>CODENAME BLASTER</i>)	10
SW-7	11
BIRD CALL HEURISTIC-BASED IDENTIFICATION AND RECOGNITION PROGRAM (B-CHIRP)	11
CATEGORY: IMAGE PROCESSING	12
SW-8	12
FLORA – GPU ACCELERATION OF PATTERN MATCHING ALGORITHMS	12
SW-9 (R)	13
COUNTING SHEEP AND CROP HEALTH IDENTIFICATION USING IMAGE PROCESSING AND OBJECT DETECTION	13
SW-10	14
SMART-PHONE INDOOR NAVIGATION SYSTEM USING EDGE DETECTION	14
CATEGORY: MECHATRONICS	15
SW-11	15
LURCCI – LEVEL-FLOOR UNDERCOVER ROBOT CRAWLER CONTROLLED BY INTERNET (PHASE 1)	15
SW-12	16
MTG - MOBILE TUNE GRIP	16
CATEGORY: HIGH PERFORMANCE COMPUTING & NETWORKING	17
LEGACY TOPICS – ASK IF YOU WANT MORE INFORMATION	17
SW-L1	17
RWTM - RHINO WIRELESS TRANSCEIVER MODULE	17
SW-L2	18


CATEGORY: SOFTWARE DEVELOPMENT & BIG DATA

<p>ID:</p>	<p>SW-1 (R)</p> <p style="text-align: right; background-color: yellow;">[project reserved]</p>		
<p>TITLE:</p>	<p>DSSAF web interface and streaming client (a Big Data theme project)</p>		
<p>DESCRIPTION:</p>	<p>This project concerns development that contributes to the Digital Signal Storage and Access Framework (DSSAF) aimed to facilitate access to recorded radar data and experiment logs. This system is planned to eventually support both intranet and internet data sharing options, but for an initial prototype it can all work in a private LAN.</p> <div style="text-align: center;">  </div> <p style="text-align: center;">Figure 1: Overview of the large-scale view on the DSSAF system.</p> <p>The RRSF has multiple on-going projects involving acquisition of radar and other types of digitized signal data. All together our field experiments constitute many terabytes of valuable recorded data, and much of this data we need to keep and make use of later as it is essential for understanding application complexities and useful for laboratory-based testing and optimizing.</p> <p>This project concerns development that contributes to the Radar Data Storage and Access Framework (RDSAF) aimed to facilitate access to recorded radar data and experiment logs. While this project is designed around being done in collaboration with a student working on project, this project is designed so that if SW-1 doesn't work out this project</p> <p>The RRSF has multiple on-going projects involving acquisition of radar data. Our field experiments generally provide many terabytes of valuable recorded data, essential for understanding application complexities and useful for laboratory-based testing and optimizing.</p> <p>This project focuses on developing a database design, setting up the database and providing an API with a set of commands for searching particular fields and getting access to data via a UDP- and/or TCP-based</p>		


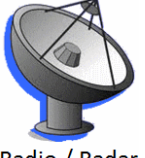
	<p>server system. A clear set of API commands needs to be design for this service and planned to be scalable / extendable. An added extra, if time permits is to incorporate a website front-end (i.e. it could use server scripts to call API functions) to make it easier to access the stored data.</p> <p>Most of our existing datasets are loosely organized (e.g., a simple file system, according to date) and it takes time to find and access this data. If SW-1 is taken, then the design of accessing this data should be done collaboratively; alternatively (or as a backup policy) this project can simply work according to file directory listings. The RDSAF is consequently an experiment attempt to improve access to this data.</p> <p>This project is designed around being integrated with SW-2, the RFSAF web client and streaming client API. However, this project is designed so that effective work can be done on it even if SW-2 is not allocated.</p>
DELIVERABLES:	<p>Database set up with a selection of radar data and information related to this, with C code for showing how to search and access data. (Possible extra: Web-based GUI for easier access).</p>
SKILLS/REQUIREMENTS:	<p>Programming, some signal processing knowledge advantageous</p>
EXTRA INFORMATION:	<p>Project Reserved – Mr. Oliver Powell</p>
AREA:	<p>Software development, Programming, Databases</p>



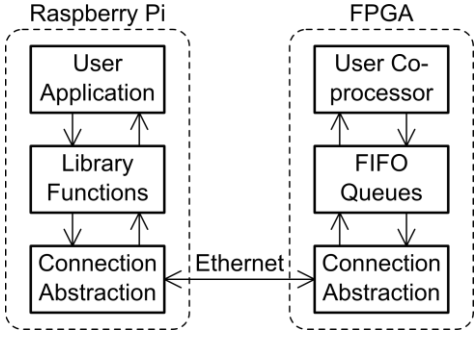
ID:	SW-2 (R) [project reserved]	
TITLE:	Tutor Organization and Planning App (TOPA) A small-scale social networking (SSN) theme	
DESCRIPTION:	<p>The Tutor Organization and Planning App is intended as a facility to get tutoring teams more organized, to help tutors get access to tutorial info (e.g., memos students doing the prac can't access) and useful course material. The system also aims to incorporate a social networking dimension, allowing tutors to be better aware of other tutors and to share tutoring expertise and resources. There should be support for formal (course-based) Tutor Teams as well as Tutor Buddies (i.e. like friend connections on Facebook). There are different access levels; only tutors in a tutor team get access to memos for the particular course they are signed up to. Generally, anyone can access the course material, even tutors in different teams (but they would need to search for this or be sent an invite). For this project the course information could just be a URL to the course website (later additions could be access to a shared folder). Support and security for memos need to be in place. Some kind of instant messaging would be desired, and means to send Team Invites and Tutor Buddy invites. Also a Lecturer user type is needed, which could just be the same as a Tutor plus permission to create a Tutor Team and to add memos.</p> <p>If the 'small social networking' functioning adequately the application can be expanded to provide for facilities such as adding a Chief Tutor user type that is able to create subgroups and to do tutor attendance lists. A "Call Tutor" function could be added to tell a tutor he/she is late for the prac, or "Request Stand-in" to broadcast a message, perhaps to the whole tutor team or to all tutors in the network to indicate that a stand-in tutor is needed (e.g. call on a tutor assigned to another course who knows the material to help out).</p> <p>There is flexibility in terms of how the system is implemented – it could be as a single web service that uses PHP/CGI/other scripts, or it could be a client-server architecture with the client being a smart phone app that talks (e.g. via a socket) to the server.</p>	
DELIVERABLES:	Operational TOPA application, short user manual helping to explain how to set up and use the system. Testing in a selection of tutoring environment (e.g., in a prac, in a tutorial, in marking session).	
SKILLS/REQUIREMENTS:	Networking & Programming (Your choice, e.g. Python, Java or C++)	
EXTRA INFORMATION:	Project Reserved – Ms. Subha Singh Can be done as a PC-based system if it is taking too long to get things working on the mobile device. Various desirable functions (if time permits, otherwise this is just a placeholder for ideas): marking, collaborating with other tutors, call tutor, schedule of who is tutoring what lab/tut when, view tutoring	


	<p>timetable and venues, confirm availability, request replacement tutor, running late, attendance record for logging students present in pracs, log of student tasks completed, expertise areas (which tutor has volunteered to focus on what difficult aspect of the tutorial), for recording test results, view sample solution. Alert lecturer: of aspect students are struggling with, of poor attendance, of plagiarism, of disturbances, of software/lab problems, of error in question / in sample solution. Other tutor admin things: log tutoring tasks, time spent on tasks, generate timesheet. Of course if you think of more items you can add that in; I'm not expecting all the myriad set of functions to be implemented, that is part of the requirements analysis and prioritization process – ideally the requirements should be gathered in a formal and thorough manner, interviewing lecturers and students, proposing your initial plans, asking for their opinions of your ideas and suggestions for further features.</p>
AREA:	Network Software

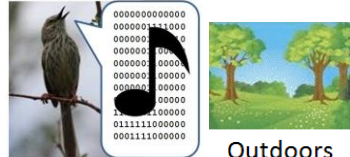
ID:	SW-3	 Networking
TITLE:	PTNS – Portable Trading Network System	
DESCRIPTION:	<p>This project is designed around supporting outdoor markets, street markets, tradeshows and other trade contexts in which a networked system is desired but the infrastructure is not necessarily in place. The system is planned primarily around providing very low cost and temporary wireless or wired networking between Point-Of-Sale (POS) terminals, Inventory Management (IM) terminals, and the Store Manager (SM) or proprietor’s terminal used for monitoring transactions and sales volumes.</p> <p>This system is intended to account for rural African conditions where there may be unreliable (if any) 3G or other high-speed networks available, or where the electricity supply is unreliable that could cause loss of sales if mains-powered equipment is needed.</p> <p>A proposed solution, the point of this project, is to establish effective principles to quickly replicate or more an ad-hoc wireless network (or one using smartphone act as a router) or ad-hoc wired network (or using a PC as a router) from one place to another quickly. The emphasis needs to be on <u>low-cost and quick</u> solutions designed around a portable network (getting the same network moved or set up quickly in a different location).</p> <p>This project is a combination of a networking and Android / Smartphone App development project. Thorough design is needed for both the software and networking aspects.</p> <p>An Android applications needs to be prototyped to respectively provide: 1) POS functionality, 2) Inventory Management (updating a database for sales made, stock added), and 3) Store Manager view (could just be database queries or if time permits a GUI app to generate reports).</p> <p>Researching the potential performance of the proposed network needs to be done, including modelling of the network conditions and computing network performance characteristics.</p>	
DELIVERABLES:	High-level Design of the PTNS; Software Design of the applications (note scope for the trial application is small scale); Prototype implementations of the POS, IM, and SM applications; network modelling & estimated performance when scaled up.	
SKILLS/REQUIREMENTS:	Software Design, Programming, Networking	
EXTRA INFORMATION:		
AREA:	Software / Networks	

CATEGORY: EMBEDDED SYSTEMS, FPGA, SIGNAL PROCESSING



ID:	SW-4	 
TITLE:	FERS Simulator GUI Upgrade (a music theme project)	
DESCRIPTION:	<p>The Flexible Extensible Radar Simulator (FERS) was developed as a PhD project in the RRSg group. This is a program and customizable coding framework that is useful both as a learning tool, for new users to the field of radar to improve their understanding of how radar and radar signals work, as well as a useful system for modelling experimental scenarios to gain better insight into the expected operation of a system, or to develop a better experiment design before going out into the field to do the experiment. The application is open-source, and available at http://sourceforge.net/projects/fers/.</p> <p>Currently, FERS is a console-based application in which a configuration file is edited in a text editor in order to describe the model that is going to be simulated. This project involves designing a GUI for FERS and integrating this with the current code implementation, and hopefully releasing it back to the sourceforge repository. This project does not require a thorough understanding of radar and signal theory, but this would clearly be beneficial; these factors can also be learned 'on the job' while working on the project.</p> <p><u>Acknowledgement:</u> FERS was originally developed by Dr. Marc Brooker.</p>	
DELIVERABLES:	Prototyped RMB device. Demonstrate sending and receiving of morsecode signals. Report on distance tests (part of thesis).	
SKILLS/REQUIREMENTS:	Programming / prior GUI experience desirable but not essential	
EXTRA INFORMATION:	Working on this project will get you well connected with the RRSg team and support will be available both from a variety of individuals, in particular Dr. Craig Tong will be available to provide additional guidelines and suggestions for GUI refinement from the perspective of an experienced user.	
AREA:	Software Development	

ID:	SW-5	 Embedded
TITLE:	Ethernet Interconnect Abstraction between Raspberry Pi and FPGA Coprocessor (<i>codename EIARF</i>)	
DESCRIPTION:	<p>This project involves connecting a Raspberry Pi (RPI) that has an ARM CPU to a Nexys3 (Spartan 6 FPGA) by means of Ethernet. The idea is to make use of raw Ethernet frames over a 100 Mbit/s connection to provide the equivalent of a high-speed peripheral bus between the RPi and the Nexys3. The project also involves the implementation of a simple co-processing application in order to test the capabilities of this connection. A model of the connection and software abstraction layers is provided in figure 2 below.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div data-bbox="475 712 911 1048">  <p style="text-align: center;">Figure 1: The Raspberry Pi platform</p> </div> <div data-bbox="981 712 1457 1048">  <p style="text-align: center;">Figure 2: Illustration of</p> </div> </div> <p>The connection itself must be implemented as an abstraction. As far as any potential application is concerned, the RPi side must be able to make use of library functions that are similar to file IO operations. The FPGA side must look like FIFO queues of incoming and outgoing data. The Ethernet layer must be completely hidden to both sides of applications using this framework. The content of the data is irrelevant, as it is application-specific.</p> <p>The HDL code (FPGA-side), as well as the C/C++ code (RPI side) must be designed around portability to larger systems, with more powerful FPGAs, faster Ethernet and higher-end ARM processors.</p> <p>This project will be jointly supervised by John-Philip Taylor and Simon Winberg.</p>	
DELIVERABLES:	<p>The project is expected to deliver a working and scalable general-purpose Ethernet connection, as described above.</p> <p>Full documentation, including a user manual, for anybody who wants to make use of the connection code, must also be delivered.</p>	
SKILLS/REQUIREMENTS:	<p>Embedded Linux development; FPGA HDL Programming.</p>	
EXTRA INFORMATION:	<p>http://www.raspberrypi.org/ : Information about the ARM board http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,789&Prod=NEXYS3 : the Nexys3 product description</p>	
AREA:	<p>Embedded Computer Engineering, Digital Systems and Communication Layer Abstraction.</p>	

ID:	SW-6	 Embedded
TITLE:	FPGA Data Blaster Kit (<i>codename Blaster</i>)	
DESCRIPTION:	<p>If you're a high-speed fundi and you're fascinated by maximizing computer performance and adventurous enough do things like overclocking CPUs to boost their performance, then this is probably a project for you!</p> <p>This is an FPGA based project which involves getting data in to and out of the FPGA at the fastest possible rate. The data input can be done in various ways (you need only implement one method), the preference being to use one of our high-speed ADC cards (can sample over 100MSps) to stream in the data from the ADC, storing it briefly into a buffer on the FPGA board, do minimal packetizing and push the data out to a multicore PC over a directly linked Ethernet cable. The PC receiving the data has a high speed Ethernet port and saves the data into memory and optionally to hard-drive (this save to drive feature can be activated/deactivated by the user depending on processing needs). If during the project the ADC is found to be too complicated to use, then a digital data source (e.g. embedded platform sending data bits to a programmable IO port) can be substituted for the ADC.</p> <div data-bbox="517 920 1425 1099" data-label="Diagram"> <pre> graph LR ADC[High-speed ADC] --> FPGA subgraph FPGA BR[Blaster Receiver] BPE[Blaster Protocol Encoder] end BPE --> Eth((Eth)) subgraph PC BPD[Blaster Protocol Decoder] RxApp[RxApp] end Eth --> BPD BPD --> RxApp </pre> </div> <p style="text-align: center;">Overview of the design for the Data Blaster Kit</p> <p>The comms does not need to use standard Ethernet protocols; instead the Ethernet card can just be used as a digital bit-stream acquisition source, i.e. override all the protocol layers and just grabbing input from the card's RX ports directly (existing Ethernet drivers can be repurposed for this or use of WireShark).</p> <p>This project requires a FPGA implementation that receives and sends data, as well as a small PC-based application that receives the data (the PC program can be very bare-bones, just dumps data to file or confirms that e.g. a sequence counting 0-1000 back to 0 is being received).</p>	
DELIVERABLES:	<p>The project is expected to deliver a working Data Baster Kit that for the FPGA platform used.</p> <p>Comprehensive documentation and tutorial on how to configure and setup the system. Performance testing methods and record of results you achieved.</p>	
SKILLS/REQUIREMENTS:	<p>Embedded Linux development; FPGA HDL Programming.</p>	
EXTRA INFORMATION:		
AREA:	<p>Digital Systems and Communication Layer Abstraction.</p>	

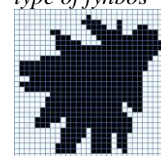
ID:	SW-7	
TITLE:	Bird Call Heuristic-based Identification and Recognition Program (B-CHIRP) <i>An outdoor theme project</i>	
DESCRIPTION:	<p>This project is aimed to be an initial pathfinder to experiment with the prospect of developing a smartphone app that could either automatically identify a bird based on recording its song, or to assist a human in identifying the bird by providing cues and playing back recordings of bird sounds.</p> <p>This project will mainly focus on algorithm development and testing, collecting a variety of bird songs (of local birds) at different volumes and distortions, and then experimenting with filters and audio pattern matching methods, or possibly machine learning / neural networks to attempt to autonomously decide a type of bird based on the sounds it makes.</p> <p>At this stage, a sophisticated user interface and comprehensive software structure is not required, the main purpose is to experiment with algorithms and establish prototype code that can later be refined. Decent module interfaces and comments in the code are desired to make the effort usable in a future application.</p>	
DELIVERABLES:	Application running on PC (or on mobile platform/smartphone)	
SKILLS/REQUIREMENTS:	Your choice of language; e.g. Matlab, Julia, Octave, PC-based Java program, or Java Mobile development using Android SDK.	
EXTRA INFORMATION:	Some existing applications that help birders identify birds: http://www.whatbird.com/ http://appcrawlr.com/android/merlin-bird-id-by-cornell-lab	
AREA:	Signal processing (optional: Embedded/Mobile app development)	

CATEGORY: IMAGE PROCESSING

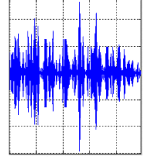
ID:	SW-8	  Outdoors
TITLE:	FLORA – GPU Acceleration of pattern matching algorithms (an outdoors & acceleration theme project)	
DESCRIPTION:	<p>This project relates to the Fynbos Leaf-based Optical Recognition Application (FLORA), which is a system that aims to automatically recognize a variety of fynbos plans based on leaf photographs. The project proposed relates to a relatively small, but critical, component of the FLORA system. FLORA is currently written in mostly Python code with some C modules for speed optimization.</p> <p>This project involves two things, where point 1 is higher priority and point 2 lower priority if time permits:</p> <ol style="list-style-type: none"> 1. Covert existing image processing operations for pre-processing stages of edge detection, binarization, and others into GPU (e.g. CUDA or OpenCV/OpenCL) equivalents and integrate with the existing FLORA coding framework. 2. Implement leaf characterization heuristic on GPU for faster generation of leaf characteristic code sequences (LCCSs); these are like fingerprints codes for branches and leaves of a particular bush which are very similar among plants for the same species. <p>The scope of this project is flexible according to time constraints. For instance the implementation can be tested with a relatively small set of samples which can be easily found around campus. This project is most suited for a student with strong programming ability, and also willing to spend a bit of time in the outdoors looking for and photographing plant specimens. A certain amount of literature review will be needed to become familiar with commonly used image-processing techniques needed for pattern matching algorithms.</p>	
DELIVERABLES:	Neat code modules that are / can easily be incorporated into the FLORA code framework. Documentation for the code.	
SKILLS/REQUIREMENTS:	Strong Programming Ability (should know some Python; you can learn how to program GPUs on the job)	
EXTRA INFORMATION:	Paper providing more detail about the FLORA application: http://dx.doi.org/10.1109/NCVPRIPG.2013.6776220	
AREA:	Image Processing, GPU-based application acceleration	

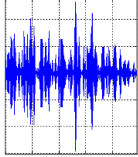


Geelmagriet a type of fynbos


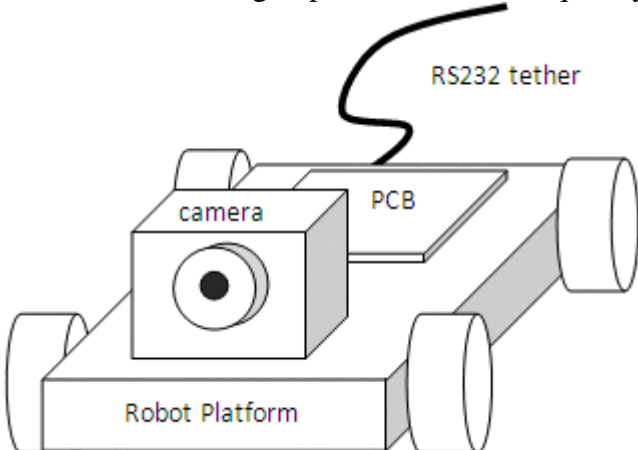



Binarized leaf photo

ID:	SW-9 (R) [project reserved]	 <p>Signal Processing</p>
TITLE:	Counting Sheep and Crop Health Identification Using Image Processing and Object Detection (an image processing and object detection themed project)	
DESCRIPTION:	<p>This project uses automated image processing and sensory data to:</p> <ul style="list-style-type: none"> Count the number of sheep in an image (this could be extended, depending of time constraints, to use machine learning/object detection, to count any specific type of animal in the image). Identify and differentiate the health statuses of different types of crops/weeds. <p>Being able to perform automated counting accurately is an active field of research which holds promise in the development of many areas. One such application relates to agriculture. Many farmers, for example, have problems keeping track of the numbers of various types of cattle. The cattle numbers are influenced by things such as the rate of births and deaths (by natural means, sickness or predators). The health status of crops is also a concern for farmers. Crops can suffer from disease and predation throughout the year and the loss of crop yield could be detrimental not only to the farmer but to the economy that depends on the revenue generated from to a fruitful harvest. This project aims to develop a computer-based means to help farmers with these issues.</p> <p>The general idea behind this project is that a UAV can capture sensory data by flying over the terrain of a farm, taking hundreds or thousands of images and stitching these together to form a map. Image processing techniques and object detection will then be applied to the areas of the map containing sections of interest (i.e. portions of the map containing cattle and crops) to count the number of sheep in the image as well as inspect the health status of various crops in the image.</p> <p>Preferably, the programs should be designed such that the necessary tasks are executed automatically upon receiving the map from a UAV.</p>	
DELIVERABLES:	Application running on a PC; or code modules that can be incorporated with a UAV. Documentation for the code.	
SKILLS/REQUIREMENTS:	Programming, image processing, object detection	
EXTRA INFORMATION:	Project Reserved – Marchello Meiring	
AREA:	Image processing, object detection, programming.	

ID:	SW-10	
TITLE:	Smart-phone indoor navigation system using edge detection (a signal processing theme)	
DESCRIPTION:	<p>The inability to use GPS in indoor environments, poses a navigation problem for users of such environments. Two common examples are; shoppers in a shopping mall, and people in a university campus. Commonly handheld devices cannot utilise GPS for localisation and navigation.</p> <p>The challenge at hand is to design and build a system that utilises edge detection to help a user navigate through a built environment using their smart phone.</p> <p>The system should be in form of client-server architecture with the client being a smart phone, and the server being the system storing the main database.</p>	
DELIVERABLES:	A client-server prototype to establish feasibility of concept	
SKILLS/REQUIREMENTS:	Digital Signal processing, Networking, Systems design, Programming	
EXTRA INFORMATION:	Can be done as a PC-based program. Preferably will use android smart-phone with a duo-core or better so as to make it mobile. An attempt was made on this last year, but I have a much approach on how it could be done better and quite differently to previously.	
AREA:	Digital Signal Processing	

CATEGORY: MECHATRONICS

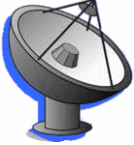
ID:	SW-11	 Mechatronics
TITLE:	LURCCI – Level-floor Undercover Robot Crawler Controlled by Internet (phase 1)	
DESCRIPTION:	<p>The plan for the LURCCI project is to develop a robot crawler that is controlled remotely via internet, intended for moving around indoors on even floors. The LURCCI robot can be built from a do-it-yourself robotics kit, which includes the chassis, and drive mechanism for moving and steering the robot – together this provides a platform on which further functionality is built. Other features would be needed to make the robot more useful. This project is design around multiple phases, each phase could be done by a different student working in parallel or in subsequent years.</p> <div style="text-align: center;">  </div> <p>This is the first time this project is being offered, and consequently, phases 1 and 2 are the requirements for this edition of the project. However, the full set of phases for the robot can be sent to you on request.</p> <p>Phase 1: Assembling the robot kit, getting the platform together and tested. Control movement of platform via a RS2332 tether. Testing speed, manoeuvrability, battery life.</p> <p>Phase 2: Vision sensor. Adding 1x camera to front of robot which will be used in later projects to enable remote control. A webcam for the first prototype or a small COTS CCD camera kit can be used (ideally this camera should be hooked up to a microcontroller running on the robot, but due to time constrains this may need to be done by a future student working taking the project further).</p>	
DELIVERABLES:	Prototyped LURCII able to move around via manual PC control	
SKILLS/REQUIREMENTS:	Mechatronics, programming	
EXTRA INFORMATION:		
AREA:	Mechatronics	


ID:	SW-12	 Music
TITLE:	MTG - Mobile Tune Grip <i>(a music theme project)</i>	
DESCRIPTION:	<p>Are you interested in making music? You sometimes think of good beats and want to tap it out? You'd like to record these tune but would like something more than a table top to test things on. But you also want something that's more natural-feeling and ergonomic compared to the flat slate of a cellphone or tablet? If this is something you'd like then maybe you'd consider taking this project on as a DIY solution for this idea.</p> <p>The objective of the <i>Mobile Tune Grip</i> is to develop a custom music input pad (the 'gripper'), on which you can play and record tunes. You can then hook it up to a computer to download and play back the tunes. The diagram below gives a notion of what is involved. You don't need to know how to play a musical instrument or how to read musical; but this project might be an opportunity for you to understand more about how music is structured and composed.</p> <div data-bbox="730 1070 1257 1406" data-label="Image"> <p style="text-align: center;">Speaker / earphones</p> </div> <p style="text-align: center;">Figure: Concept sketch of MTG</p>	
DELIVERABLES:	The gripper and electronics needed to produce and capture music. Optional extra: integration with PC to download recorded tunes.	
SKILLS/REQUIREMENTS:	Electronics, microcontroller programming, C programming	
EXTRA INFORMATION:		
AREA:	Electronics, Embedded systems	

CATEGORY: HIGH PERFORMANCE COMPUTING & NETWORKING

None offered due to limitations on number of project that can be listed.

Legacy topics – ask if you want more information

ID:	SW-L1	
TITLE:	RWTM - Rhino Wireless Transceiver Module (a Reconfigurable Computing theme project)	 Radio / Radar
DESCRIPTION:	The Rhino project is a reconfigurable computing platform designed for Software Defined Radio (SDR) applications. The Rhino platform was built at UCT by the SDRG research group (see the www.rhinoplatform.org Rhino site for details). The objective of the RWTM system is to develop a wireless transceiver module for the Rhino, a low-power FM transmitter/receiver that can send and received bytes of data. A speed such as 16KBps, using a simple whip antenna, may be adequate to prove the concept. It is hoped that the HDL and other code developed as a results of this project will contribute support resources and examples for the Rhino support kit.	
DELIVERABLES:	Prototyped antenna and analogue front-end (which connects to an A2D sampling module), VHDL/Verilog code for the front-end processing and Wireless communications protocol handling.	
SKILLS/REQUIREMENTS:	Embedded systems; HDL coding; digital signal processing; a reasonable understanding of signals and RF transmissions (although much of the RF skills you can pick up as part of doing this project)	
EXTRA INFORMATION:		
AREA:	Digital systems; Electronics	

ID:	SW-L2	
TITLE:	Alert Me App Coding Framework (an Android App theme project)	
DESCRIPTION:	A code framework for use with Android that can be customized for various application purposes, I'm suggesting it is tested as a kind of imminent load shedding alert. Client/server system. Client is Android application code, server is PC-based code that can send data via a socket to the client. The client side needs suitable audio and visual displays to show alerts. Different levels of alerts: low priority (maybe no sound or single buzz) to high priority (loud longer alarm, needs to be acknowledged, switches off only after e.g. a minute if not acknowledged).	
DELIVERABLES:	API user-guide. Example / starting point programs. Performance testing. Prototyped application using the API.	
SKILLS/REQUIREMENTS:	Programming. (Your choice, e.g. Python, Java or C++).	
EXTRA INFORMATION:		
AREA:	Mobile app programming / telecoms / software development	