2018 BSc Project Topics

proposed by

Dr Simon Winberg

Department of Electrical Engineering
University of Cape Town

List of Project Topics

Unallocated topics are listed first!

List of Reservations

ID:	**SW-10**	 **Embedded**
SUPERVISOR:	**Simon Winberg**	
TITLE:	**Fast Booting with Linux (industry linked!)**	
DESCRIPTION:	A frequent problem with embedded devices is the time it takes for the system to go from power on to the system becoming usable. Ideally, this time should be as low as possible. The objective of this project is to decrease the boot time of a provided system as much as possible. Factors to be investigated are: bios, boot loader, kernel, user-space init, as well as the program itself that provides the critical functionality. Thorough performance measurements and reporting needs to be provided. The various profiling and performance analysis techniques that were introduced in the EEE4084F course can be leveraged in this project in order to both provide a detailed picture of the boot process, system time and user application performance. The report does not need to be limited just to those optimization attempts that provided speedup bit also those attempts that were considered and tested but didn't necessarily provide an improvement (of course these 'deadend' attempts would need to be detailed briefly and explanations of why they failed; the main results should of course focus on attempts that did provide useful benefit in improving the speed).	
DELIVERABLES:	Detailed report on the 'optimization journey' of the faster Linux booting approach (covering both the 'slips' and 'ascents' of the journey). The student doing this project is expected to report regularly to the supervisor (S. Winberg) and to provide regular fortnightly brief progress updates to our industry partner. Two demos (one a term) are planned to show the industry partner concrete results of progress that is being made.	
SKILLS/ REQUIREMENTS:	Embedded Systems / Software Engineering	
ELO3: Engineering Design	This project needs to provide a thorough design of the problem to be optimized, showing a clear view on the different levels of the boot process, the types of initial operations that need to be completed and the processes to be started prior to expectation of the main program. The student needs to thoroughly investigate the various aspects of the Linux OS that could be configured or tweaked (in code or via configuration settings) to achieve a faster boot. Thorough performance evaluation is to be performed and reported on.	
AREA:	Computer Engineering	

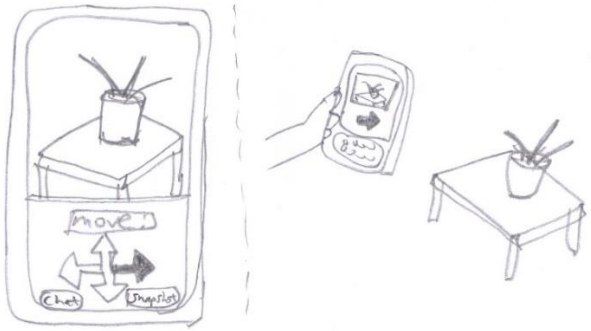
ID:	**SW-11**	 **Embedded**
SUPERVISOR:	**Simon Winberg**	
TITLE:	**Real-time Emulation Framework for Linux (industry linked!)**	
DESCRIPTION:	In many applications it is desirable to have the guarantees offered by a real-time system as well as the convenience of working in a Linux system. Although Linux isn't a real-time OS, techniques can be used to approximate one. The objective of this project is to investigate what existing facilities offered by Linux could make it behave more like a real-time system, for example using real-time priority levels, kernels space programs and dedicating cores to real-time tasks. Characterise the trade-offs of each approach utilized. From these findings, construct an application framework that will make it easier for developers to utilize your RT emulation techniques to develop their own programs. A suitable case study needs to be constructed to trail the techniques in a real-world real-time application. A suggestion is sound source localization or sonar. At least the following metrics should be considered in determining the behaviour and performance of the system: throughput, average latency, maximum latency, latency jitter, wasted resources, and possible overhead. This project is to be done in partnership with Mr Pierre Watts from UMan Technologies South Africa. An embedded platform and necessary tools will be provided.	
DELIVERABLES:	Application framework for RT Emulator Framework for Linux, in addition to sample code (case study) showing how to use the framework. Student doing this project is expected to report regularly to the supervisor (S. Winberg) and to provide regular fortnightly brief progress updates to our industry partner. Two demos (one a term) are planned to show the industry partner concrete results of progress that is being made.	
SKILLS/ REQUIREMENTS:	Embedded Systems / Software Engineering	
ELO3: Engineering Design	This project has a significant design element. The student needs to thoroughly investigate the services and features available in Linux that could be leveraged to emulate RT performance. The application framework for this project needs to be well designed, accounting for easy of reuse and configurability, in addition to facilitating performance measurements of applications built using the framework. Thorough performance evaluation is to be performed and reported.	
AREA:	Computer Engineering	

ID:	**SW-12**	 Machine Learning  Computer Graphics
SUPERVISOR:	**Simon Winberg**	
TITLE:	**Analysis of PIGSOM machine learning method for disease prediction**	
DESCRIPTION:	The Scientific Computing Research Unit (SCRU) at UCT makes use of various Machine Learning (ML) methods, such as artificial neural network (ANNs), self-organizing maps (SOMs) and deep learning to develop applications for early prediction of various types of disease from data obtained from tissue and blood samples. The focus is at present towards early prediction of different types of cancer. We have a team of students developing an application to facilitate early detection and diagnosis. We need a student to help implement additional classification and production models. In particular, the addition of a Partial-Input Growing Self Organizing Map (PIGSOM). The PIGSOM method is largely untired, it is a nontrivial machine learning method to implement as it needs to be tolerant of missing inputs. Existing prediction systems that we are using assumes a full complement of input parameters, such a complete gene sequences. This level of data is obtained from thorough, hospital-based blood tests and biopsies. But these are expensive and time-consuming procedures. Our aim to do develop a lower cost solutions that can be self-administered or performed in a rural clinic which may not have the lab facilities needed for thorough analysis of the samples. The low-cost procedures can provide only partial data compared to what the fill tests provide. Therefore we want to incorporate into the diagnostic application support for partial input – this is where the PIGSOM would be utilized. The 'growing' aspect of the PIGSOM is to allow for scalability – so that the system could learn new types of disease classification or subtypes of known diseases. This growing aspect is optional; support for partial input is the priority. Tests would include seeing at which point the classification becomes too inaccurate, i.e. the system is not able to perform effective categorization of input samples. *Optional:* If time permits it is desirable to provide a visualization for the PIGSOM results, such as a U-matrices or scatter plot. The SCRU lab has a 'vision-wall' that you could try out to present your visualizations. You could allow for additional UI and visualization controls such as setting the amount of inputs to be randomly removed, and supporting side-by-side comparison with views for results with differing amounts of missing inputs. Furthermore if you are interested in HPC or parallel programming, then you are most welcome to implement a parallelized version of PIGSOM e.g. using MPI to run on the cluster or OpenCL/CUDA for GPU acceleration. For this project you would be encouraged to work in the SCRU lab with the postgrad students and to develop your application so that it can be compiled and run on the group's high-performance server. *This topic could lead on to an MSc related to Scientific Computing / Machine Learning with the SCRU group (postgraduate funding available).*	

DELIVERABLES:	Prototyped program and well documented code. Survey from users reporting on the usefulness of the application. Experimental results showing both accuracy and performance of representative tests, e.g. comparing to basic SOM implementation.
SKILLS/REQUIREMENTS:	Software Engineering. Programming skills. (You don't need prior experience with machine learning/NN, an interest in algorithms is desirable).
ELO3: Engineering Design Perform creative, procedural and non-procedural design and synthesis of components, systems, engineering works, products or processes.	Presentation of the overall design of the system, clearly show and explain the components of the system and how the connect together. You may likely be building upon existing code/libraries so you need to explain how your contributions connect to these underlying parts. Logical testing methods to show accuracy and performance of the program. Documentation and flowchart describing the system. Software design.
EXTRA INFORMATION:	Done in partnership with SCRU: http://www.scientificcomputing.uct.ac.za/ (workspace will be provided in computing lab close to vision wall)
AREA:	Computer Engineering / Software / Machine Learning
BOOKING	

ID:	**SW-1**	 **Networking**
TITLE:	**PTNS – Portable Trading Network System**	
DESCRIPTION:	 This project is designed around supporting outdoor markets, street markets, tradeshows and other trade contexts to assist in the running and coordination of one or more trading sites. The vision for this system is to have it planned to be extensible to a much wide application domain – towards the concept of a 'Trading Cloud System' (TCS) to support shops that have multiple sites and need to manage and coordinate their staff, inventory and pricing. However, it is important to note, that this is a smaller scale start' the PTNS is designed around being a smaller start towards this larger objective, whereby there could be multiple PTNS networked systems each independently maintained and managed. The PTNS should ideally be designed around allowing the software to be reusable and customizable so that anyone can adapt it to their specific needs. The PTNS is planned around supporting only a few small stores that might be set up in a market or tradeshow. The main requirements are as follows: • Support for multiple users and access levels (e.g. manager, cashier, shop assistant) • Chat / Instance Messaging (IM) between clients • Server / master client – where the backend runs. Stores the database to keep track of stock etc. Logs chat IM. • Stock logging. Add stock / sell stock / discard stock (i.e. mark stock items as spoilt). Updates server. • POS functionality – handle selling, generate receipts (hint, see E-Receipt Cloud project the student of which you might collaborate with). Handle returns/refunds/exchanges. • Handle discounts, gift cards or store credit (e.g. on a return the user might be given credit instead of cash). • Mechanism to adjust prices, etc. (might not be accessible through POS). • Balance enquiry. Able to email the user an account balance (e.g. if they received a credit refund for a return). • Added 'Intelligence' – e.g. alters users and manager to high volume / low volume trading, for instance if one stall is very busy and another is not busy then the service sends a message to the non-busy stall to suggest sending an assistant to help.	

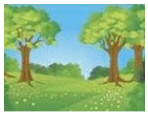
	This project is a combination of a networking and Android / Smartphone App development project. But it could be simplified down to a web-based system or to a client-server PC-based system written in e.g. Python or C++ using a GUI library like Qt. A thorough design is needed for both the software and networking aspects of this project. The client applications (whether running on Android or a plain PC application) 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). Need to also investigate the potential performance of the system, including modelling network conditions (different loading) and performance. *Potential added value:* If there is time you could attempt to design the project around supporting ad-hoc networking, e.g. two tablets connected via wifi, which would be cheaper than attempting to use a mobile network for data or having a purchase wifi credit. *NB:* student working on this project could potentially collaborate with the student working on the e-receipt cloud project.
DELIVERABLES:	High-level Design of the PTCS; 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
ELO3: Engineering Design	*The project will involve a significant amount of design, implementation, usability analysis and evaluation of the software system as per the requirements.*
EXTRA INFORMATION:	*https://www.softwareadvice.com/resources/what-is-a-point-of-sale-system/* https://chubbable.com/things-you-need-to-know-in-building-your-own-pos-system
AREA:	Software / Networks
RESERVATION:	*Reserved Mustafa Rashid* rshmus001@myuct.ac.za (speak to lecture if you want to do something similar)

ID:	**SW-2**	Smartphone Application
SUPERVISOR:	**Simon Winberg**	
TITLE:	**Show Me Around App (SMAA)**	
DESCRIPTION:	This project is about developing a system whereby two smartphones can be linked, where the one can be used to show user of the other what a particular environment or view looks like. Imagine that you're standing in a room that has a whole lot of fascinating displays and you want to share this with a friend right away… but how can you? You could take some pictures, go into WhatsApp and select the various pictures and send them on. But that's so slow. You could take a video, save it somewhere or upload it to Google Drive and share the link – that's bit of a hassle too and not interactive. No, it would be so much better if you could just start an app and move the phone about to show the scene, and have someway for your friend to direct what you are showing, e.g. if you friend wanted to look at something specific they could just press buttons or click on something they want you to focus on or zoom in on. So that's essentially the purpose of the ShowMeAround App. The concept sketch below gives a suggestion of how the system could work.  Viewer's phone on the left, and on the right the Shower is holding the Shower's phone panning the environment of interest. There's two users: the Viewer and the Shower. The Shower is in the envorinonment being shown and moving around the shartphone that is recording the images. The Viewer's smartphone is used for remotely seeing what the Shower's smartphone is recording. The system could potentially record both sound and images, but the main objective is just for visuals. There could be two modes: Cursor Mode where the Viewer presses arrow keys to tell the Shower how to move his/her phone around. In Point Mode the Viewer taps an object in the scene that she Shower needs to focus in on (in this mode the Shower obviously also needs to see what images the phone is presenting to the Shower in order to see what the Shower is tapping). Advanced option: If you feel particularly inspired and have time available you could attempt to extend this to a 3D remote modelling app, where the Shower pans the phone and software such as SLAM is used to generate a 3D model that is sent to the Viewer to explore.	
DELIVERABLES:	Operational Show Me Around Application, that should allow at least the Viewer side to be running on a smartphone. The Shower side could run on a PC with a webcam used to record the scene.	

SKILLS/ REQUIREMENTS:	Android and or C++/Python/Java programming.
ELO3: Engineering Design	*Thorough design of this client-server system. Provide a system level design showing clear distinction of the Viewer and Shower sides of the system. Detailed design of the separate Viewer and Shower subsystem. Good documentation of the application. Thorough evaluation and performance analysis of the latency and responsiveness of the system.*
AREA:	Computer Engineering
RESERVATION:	*Reserved: Tatenda Muvhu MVHADM001@myuct.ac.za*

ID:	**SW-3**	 **Embedded**
SUPERVISOR:	**Simon Winberg**	
TITLE:	**ComfyRide App**	
DESCRIPTION:	 This project concerns developing an electronic analysis device that can be utilized to measure the comfort of riding in a vehicle and the comfort associated with a particular road or section of road and the noise levels in terms of tire nose along the route. The device measures the intensity of bumps or jolts and the timing between significant jolts. For instance, the roads connecting a starting point A and ending point B might on average be along a not very smooth tar roadway that causes a lot of tire noise but might have a low level of bumps (e.g. potholes, corrugations or other types of uneven sections). The system needs to provide reports of the comfort of 1) a particular car, 2) particular routs (from a point A to point B) which could be names e.g. "home to work", and 3) if there is time for more advanced features: the system could connect to GPS and a real-time map service (e.g. Google Maps) to record and link comfort information to specific roads. The version of the application developed in this project can be considered a starting point, depending on how for you get, to the ultimate purpose. The ultimate purpose to provide a 'Ways' app type of service that will use crowd sourcing to gather statistics about particular roads and the use this information to e.g. plot the most comfortable route (as opposed to the fastest or most direct route), and as a means to feed information to municipalities about the quality of their roads and which sections might be due for maintenance.	
DELIVERABLES:	Functional ComfyRide app that can record conform information of a car (or other form of transport, e.g. train) journey.	
SKILLS/ REQUIREMENTS:	Embedded systems, Android development desirable (but not essential, can implement on a laptop connected to a remote device for measuring vibrations etc.)	
ELO3: Engineering Design	The system design, showing the integration of the ComfyRide system including high level design, low level design, user interface and communication, as well as clear explanations of integration aspects of the various parts that may be developed.	
AREA:	Computer Engineering	
RESERVATION:	*Reserved: Andrew Oliver (OLVAND008@myuct.ac.za)*	

ID:	**SW-4**	 **Embedded**
SUPERVISOR:	**Simon Winberg & Prof. Michael Inggs**	
TITLE:	**Adjustable precision processor and computation cost analysis**	
DESCRIPTION:	The objective of this project is to construct a VHDL or Verilog framework by which to experiment with performing common signal processing operations at variable levels of precision. This project is based on research that has been started by John Collins on the topic of investigating the numerical precisions required to execute real world programs. The aim of this project is to compare the use of logic, electrical power usage and processing speed for a selection of processing operations using different levels of precision. Standard 32 bit floating point or fixed point numbers potentially provide more precision than what is needed, meaning more data is being stored and handled than necessary; and the extraneous bit switching that results can cause the system to utilize more power than necessary, as well as possibly taking longer to complete calculations (e.g. managing bit carries and transferring data etc.). This project sets out to measure costs of computation for a selection of processing operations. The plan is to first implement trial algorithms, running at full (32 or 64 bit) precision on a PC to check the calculations are correct and to establish a golden measure. Then implement the operations in VHDL/Verilog to run on an FPGA. Provide calculations (such as +, -, *, / to work for varying levels of precision and size, e.g. 8 bit, 12 bit, 16 bit, 24 bit and 32 bit floats). Compile the designs to see changes in compile (trace & route) times, logic elements used, maximum clock speed, etc. and report on these differences. The calculations can then be run on a simulator to investigate the results, how well they match to the golden measure. Further tests can be done to see at what point the computations break down due to too little precision. Then (if time permits) adjust the HDL code to run on an FPGA platform (e.g. a Rhino Platform what has power meters on board that can provide precise power usage measurement) – if it is not sufficient time to run on physical FPGA board then accurate estimates for the power usage can be calculated based on datasheet information and aspects such as number of bit inversions during processing (which can be calculated by saving simulation results to a file and writing a script to count the bit flips for each register/port logged. This project has much research potential and real-world application, which can make a useful contribution to the field of computing.	
DELIVERABLES:	HDL-based designs of selected signal processing operations (can have some simple operations such as peak detector, averaging filter, LPF, and if time permits more complex filters such as FIR filter but this is not a requirement for this BSc level project)	
SKILLS/ REQUIREMENTS:	Verilog / VHDL coding and C/C++. (Fortran programming beneficial).	
ELO3: Engineering Design	The project will involve designs, implementation/ simulation, analysis and evaluation of the system as per the requirements.	
AREA:	Computer Engineering	
RESERVATION	*Reserved: Keegan Crankshaw CRNKEE002@myuct.ac.za*	

ID:	**SW-5**	 Outdoors Signal Processing
SUPERVISOR:	**Simon Winberg**	
TITLE:	**Isolating a moving object of interest from video frames that may exhibit background motion.**	
DESCRIPTION:	Motion draws the attention of an observer. For example when someone is waiving a flag at the roadside your attention is likely drawn to that motion, which could be indicative of a potential change to the environment, which could possibly influence you. Moving objects may exhibit different types of motion: they may be simply vibrating or moving-back-and-forth quickly or slowly, which might have little influence on the observer. For example tree branches and leaves may be moving in the wind, or waves in the sea may be moving in a scene. This project involves isolating moving objects of interest from the rest of a scene that may comprise stationary and moving objects that are not significant. Experimental techniques will be trialled to extract moving objects of interest from a series of frames. Ultimately the goal for this processing would be to have the methods incorporated into a system that can process large amounts of video sequences and propose which moving objects may have caused an event in the environment monitored at a certain time – however the scope of this project is focused on extracting moving objects of interest from a series of images, the two figures below shows an example of what the application would do. The motion isolation system needs to be designed around fitting into a higher level application that may contain it; i.e. you are to develop a subsystem or software module that is designed around fitting in to a larger software application – in order to do this, a trial larger-scale application is proposed as a motion logger application (MLA). The MLA calls on the underlying motion isolator function to find the objective of interest and take a snapshot of it, recording it in an image file. The snapshots can then be played back showing what objects exhibiting motion were found a series of frames. Image processing techniques such as blurring (to eliminate small changes) together with Otsu's thresholding and Connected Component Labelling (CCL) to find a single, connected region (i.e. to separate a potential object of interest from the background). The application can be expanded further to utilize machine learning techniques to identify or classify the extracted object(s) in motion based on a database of learned shapes.	

	
Object in motion (seagull) upon vibrating background (sea waves)	Object of interest (seagull) in motion isolated from scene

DELIVERABLES:	Prototyped program and well documented code. Experimental results, showing both accuracy (how well the object was extracted and if the extracted parts were all from the object of interest), quality (does the system perform consistency well) and performance (particularly speed) of the application. Example application where motion isolation feature is used as a component in this higher level application.
SKILLS/REQUIREMENTS:	Programming skills. Some experience in image processing is beneficial but is not a requirement as this project provides an opportunity for the student to develop these skills during the project.
ELO3: Engineering Design Perform creative, procedural and non-procedural design and synthesis of components, systems, engineering works, products or processes.	Documentation and flowchart describing the moving object extraction techniques and resultant algorithms applied. Software design and well documented code for the prototyped program for extracting moving objects of interest from video frames.
EXTRA INFORMATION:	
AREA:	Programming. Image processing.
RESERVATION:	*Reserved: Claude Betz BTZCLA001@myuct.ac.za*

ID:	**SW-6**	 **WebService**
SUPERVISOR:	**Simon Winberg**	
TITLE:	**UniProjs**	
DESCRIPTION:	 So many good university projects disappear, or starts on grand schemes are made and then forgotten. But the *UniProjs* concept is a proposal to provide a means to extend the life and sustainability of university projects – and at the same time help students and academics more easily deal with project management, backups and supervision. UniProjs is both a means to keep track of projects, allowing students to develop and submit gantts charts and other planning documents, parts lists, schedule meetings among other project related activities. But it also allows a way to record project resources and a means to archive completed project code and other artefacts. But you might say "but that's crazy there's surely lots of free software out there already!"… well there is, but none of it is totally suited to the way we do things for the BSc projects or for postgraduate projects. If you compare the project management strategies and delivery of forms and progress tracking documents of a typical BSc project to something that a commonly use tool like Redmine provides there are a log of gaps in terms of needed functionality. But besides all that wouldn't it be quite a thrill if future BSc students and postgrads were using a project management and repository system that you started! The UniProjs is planned to leverage a certain amount of available open-source tools in order to accomplish its objectives, for instance Redmine could be repurposed and customized. A git repository could be setup and template project folders constructed. This type of project is expected to provide the student doing it with both excellent programming skills (customizing code and possibly developing their own online services) as well as developing good insight into project management for electrical / computer engineering projects that could provide a rather impressive project feature in a CV in demonstrating your deep understanding in organizing and managing engineering projects, a skill that is highly desirable in the workplace.	
DELIVERABLES:	Operational UniProjs project management system, including sample project and templates.	

SKILLS/ REQUIREMENTS:	Excellent engineering project management skills. Software engineering. Programming.
ELO3: Engineering Design	While this project needs the student to be thoroughly informed about engineering project management practices, there is also a significant amount of design work that needs to be done in order to explain how the system will be structured and the mechanisms by which computer-based facilities will be developed or adapted to provide the needed project management and project repository facilities.
AREA:	Software Engineering / Computer Engineering
RESERVATION:	*Reserved: Petrus Kambala KMBPET001@myuct.ac.za*

ID:	**SW-7**	 **Embedded**
SUPERVISOR:	**Simon Winberg & Dr Lerato Mohapi**	
TITLE:	**Integration of Smart Personal Devices for Security and Health Monitoring in Cars**	
DESCRIPTION:	This project is about the design of an integrated smart personal security and health monitoring system. This system is aimed at providing a multi-input automotive security system that cross-checks the individual's identity based on biometric information, such as the following: • Driver's car Seat-based Weight Sensor (SWS) • Visual recognition (i.e.facial) • Steering Wheel Finger printer reader (off the shelf device) When output from majority of above sensors is 'yes', that it is the right person, then the car ignition can automatically or manually start, otherwise it locks itself, while sending a message to the owner's cell phone to allow use (i.e. by entering an OTP), or if the owner wants to alert the police or security automatically. Furthermore, these sensors can be extended by introducing the following sensors for additional health monitoring: • Seatbelt heart rate monitor • Seat Angle Monitor for monitoring safe driving seat angles • Visual analysis of facial expression to determine if long distance drivers need some rest, etc. When the person's heart rate is abnormal, the system must alert the driver. When overweight and/or gaining significant amount of weight, the system must alert the user as this may affect the security knowledge-base and his health. When the facial expression analysis provide signs of sleepiness, the system must alert the driver as this may cause accidents. We of course do not expect all these features to be added, but this project can rather be a starting point. A major focus, and which could be the main deliverable due to time limits, would be the development and testing of the driver's seat-based weight measurement sensor.	
DELIVERABLES:	- Design of this smart personal system - Example devices which can be used for this systems - Security test case: Integration of Weight sensor system with facial and fi nger print recognizer using Arduino YUN board and a software based facial and fi nger print recognizer - A USB camera can be used to mimic real-time facial recognition, but finger pints can be pre-recorded.	
SKILLS/ REQUIREMENTS:	- Programming using C/C++ - Image processing techniques	
ELO3: Engineering Design	- Design of the integration system for smart personal devices in cars for security and health monitoring. - The resultant algorithms for multi-inputs smart personal security and health monitoring system - Design of a test case prototype for validation	
AREA:	Computer Engineering / Image processing / Software Engineering	
RESERVATION	*Reserved: Uveshin Moodley MDLUVE001@myuct.ac.za*	

ID:	**SW-8**	 **Embedded**
SUPERVISOR:	**Simon Winberg**	
TITLE:	**VendPkit: a cashless smartphone-based vending machine payment system**	
DESCRIPTION:	This project involves the upgrading of an existing coin-operated vending machine to make it compatible with e-payment solutions. The specific e-payment service to use is to be determined as part of the literature review for this project, it is suggested to use a service such as PayPal and its micropayments – but a PayPal micropayment of around £5 is far too much for a small chocolate or cup of coffee. So if no suitable system can be found that exists for the needs, then this system will be converted into developing a framework for micropayments, for which the proposed vending machine could provide a case study for how the system can be used. The system also needs to provide digital receipts for loading funds (e.g. if a separate microtransaction banking system is implemented) and for transaction receipts (current vending machines generally provide no receipts, so if the machine fails to deliver a product there is no receipt for the client to claim credit for the undelivered item). This project is planned around the development of a net-centric embedded system, in which the student will need to prototype the implementation of low-cost electronic component to install into the vending machine as well and the surrounding network infrastructure for the payments service (note that the network system could be very rough, more as a proof of concept than a fully usable system – a future project could focus on developing a more complete and robust the network system for this application). The proposed platform to use is a Raspberry Pi, programmed using Python. Recommendations for extending the system to (hypothetical) broader use will be an important aspect of the report, including recommendations for electronic components, pcb design and computing technologies that could be realized should this system be redesigned to use its own low-cost computing infrastructure optimized for this application.	
DELIVERABLES:	Minimum deliverable: electronic system to enable electronic payment for a vending machine, together with usermanual and technical documentation.	
SKILLS/ REQUIREMENTS:	Embedded Systems / Software Engineering	
ELO3: Engineering Design	This project has a significant design element. The student needs to thoroughly investigate e-payment microtransaction services as well as selecting appropriate software and hardware components for the embedded system to integrate with the vending machine. A thorough design for the mechatronic system, needs to be provided elaborating in particular the hardware and software aspects of the embedded system to be developed. A student pursuing this project is strongly recommended to make use of existing code libraries and other resources where possible so that the implementation can build upon	

	reliable and trusted solutions, avoiding unnecessary reinvention of solution strategies. In addition, thorough testing, robustness and reliability as well as performance evaluation is expected to be performed and reported on.
AREA:	Computer Engineering
RESERVATION:	*Reserved: Hannes Beukes BKSJOH009@myuct.ac.za*

ID:	**SW-9**	 **Embedded**
SUPERVISOR:	**Simon Winberg**	
TITLE:	**ActionTracer Sensor and Android App API**	
DESCRIPTION:	This project is about designing a small electronic sensing device that can record motion or vibration and can either log the motion internally or transmit the motion via a short wireless link to a receiver unit (a cell-phone) that will log and display information about the action. This application involves both the development of the ActionTracer Sensor and the Android code for the ActionTracer App. Each of these aspects are detailed further below. The illustration below shows how the action tracer might be used. It is important that the ActionTracer sensor be small and battery-powered, and should last for about 4h. The Android App needs to be designed around customizability, so that it can be integrated into a larger application e.g. as the diagram below suggest to log golf swings and show statistics about the swings.  As indicated in the picture, the sensor should be small enough to be connected for instance to the end of a golf club. The Android App code should be customizable to provide useful logs or displays about the activity being pursued. In this project a golf application could be implemented as a case study for using the ActionSensor for recording information about golf swings. This could be done quite simply in the lab using a metal rod to substitute for a golf club – although ideally it should be tested on a real golf club to provide input in its appreciations in such an application. If preferred an alternate activity could be tested, e.g. for driving a car and measuring the comfort of the ride (see ComfyRide App – or you could collaborate with the student doing this project if someone chooses it). Action Tracer Sensor: the sensor needs a tri-axis accelerometers to measure motion in pitch, roll and yaw. A small microcontroller, e.g. an 8-bit PIC, needs to be constantly reading the motion angles. For power-saving the system could possibly be design such the microcontroller is in sleep mode while there is no motion, and a purely analogue ‘sentry’ circuit wakes up the microcontroller when movement is detected. The user could perhaps know that the first

	motion will not be fully recorded (but for the golf app example this should be no problem because the club will be moved around for many seconds before a swing is attempted). The sensor can send out a datasteam over Bluetooth or similar low-power PAN wireless link. Action Tracer Sensor App API: at a minimum the host software side of this project just needs to receive the wireless motion data stream and record this into a file (e.g. a csv file). It is *imperative* that the provided code is designed around being a reusable API!! i.e., not a once-off application – it needs to be program code that can become the baseline for further software that can implement specific applications using the wireless link and motion log. But preferably for this project a case study should be developed showing how the API code can be reused to e.g. develop the golf swing statistics application illustrated above.
DELIVERABLES:	At least one rugged and operational ActionSensor Tracer able to send motion sensor logs wireless. API code that can receive the wirelessly transmitted motion sensor logs and provide a means to show this information (e.g. on the screen of an Android device).
SKILLS/ REQUIREMENTS:	Embedded systems, Android development desirable (but not essential, can implement the host using a PC)
ELO3: Engineering Design	The system design, showing the integration of the ActionTracer Sensor and App needs to be provided. Detailed design of sensor and app needs to be provided together with specifics about the communication protocols and mechanisms.
AREA:	Computer Engineering