

# Final Year Project 2013/2014

## General Requirements

Recommended development platform for all projects is Linux. Any distribution is acceptable, but [Slackware](#) is recommended. I also recommend projects that involve source codes (e.g. HDL, C, C) to use source code management (SCM) or revision control software such as [\[\[http://git-scm.com/ git\]\]](http://git-scm.com/) or [mercurial](#). I will be providing a wiki page for every project and I expect students under my supervision to keep a project journal within that wiki page. The template page will be given. To stress the importance of this journal, I will be making it part of your final evaluation marks. ===== General Comments ===== For each project title, I've included some keywords, objective(s) and scope. The keywords are there for you to do some background research before you jump into this - I don't want you to give-up on me (it's a waste of everybody's time) towards the end of the project. I put in objectives and scope to let you know what is expected out of this project. So, your results will be based on these expectations. I define an objective as the targeted final result. I usually set only ONE primary objective and optionally a secondary objective that will/should not affect your final results (though it usually gives you extra marks if completed) - too many objectives usually leads to unsatisfactory results. Meanwhile, scope visualizes how much work is expected - for example, one can say that the primary objective is to design a microprocessor. So, the scope should mention maybe something like you're expected to design a 4-bit microprocessor up to schematics level with instruction set that covers data transfer/processing and program control, complete with logic simulation results. ===== Final Year Project 2012/2013 ===== ===== General Requirements ===== Recommended development platform for all projects is Linux. Any distribution is acceptable, but [Slackware](#) is recommended. A good alternative would be [Debian](#). I also recommend projects that involve source codes (e.g. HDL, C, C) to use source code management (SCM) or revision control software such as [git](#) or [mercurial](#). I will be providing a wiki page for every project and I expect students under my supervision to keep a project journal within that wiki page. The template page will be given. To stress the importance of this journal, I will be making it part of your final evaluation marks.

## General Comments

For each project title, I've included some keywords, objective(s) and scope. The keywords are there for you to do some background research before you jump into this - I don't want you to give-up on me (it's a waste of everybody's time) towards the end of the project. I put in objectives and scope to let you know what is expected out of this project. So, your results will be based on these expectations.

I define an objective as the targeted final result. I usually set only ONE primary objective and optionally a secondary objective that will/should not affect your final results (though it usually gives you extra marks if completed) - too many objectives usually leads to unsatisfactory results. Meanwhile, scope visualizes how much work is expected - for example, one can say that the primary objective is to design a microprocessor. So, the scope should mention maybe something like you're expected to design a 4-bit microprocessor up to schematics level with instruction set that covers data transfer/processing and program control, complete with logic simulation results.

# Project Titles

## Development of an ARM-based Mobile Robot Controller Board

**Comment:**

This project is purely hardware development.

**Keyword(s):**

ARM, microcontrollers, mobile robots

**Pre-requisite:**

Digital Electronics, Microprocessor System

**Objective:**

1. to develop a mobile robot controller board with an ARM-based microcontroller

**Scope:**

1. produce a working prototype for mobile robot controller
2. include/enable basic mobile robot control interface (motor, servo, sensor, etc.)

## Mapping and Localization for Mobile Robot Platform Using Player/Stage

**Comment:**

This project is purely software development but with embedded systems application.

**Keyword(s):**

mapping, localization, simultaneous-LAM (  [SLAM](#) ), mobile robot, player/stage software

**Pre-requisite:**

A in Computer Programming...? (This is NOT a “Hello, World!” programming project!), Good in Maths

**Objective:**

1. to develop a practical mapping and localization algorithm/application for mobile robot platform

**Scope:**

1. using [this](#)
2. something like [this](#)
3. test on simulator platform only...

## Development of 8051 Development/Training Kit

**REVOKED!** 

**Comment:**

This project also involves a combination of hardware and software development.

**Keyword(s):**

8051 microcontroller, compilers & assemblers, software library, SDK, FPGA

**Pre-requisite:**

Microprocessor Systems, Embedded System Design...

**Objective:**

1. to develop an 8051 microcontroller development and training kit

**Scope:**

1. develop a basic 8051 soft-core to be implemented on FPGA
2. develop assembler/compiler for 8051 (optional: worst case scenario, use existing)
3. develop a software C library for various common devices (LCD, keypad, etc.)
4. design reliable testbench to verify the core operations/interface

## VGA Controller Module for Custom FPGA-based Vision Systems Verification

**NO TAKERS!** 

**Comment:**

This project is about digital circuit design on FPGA platform.

**Keyword(s):**

VGA controller, FPGA, HDL

**Pre-requisite:**

Digital Electronics, Microprocessor System (and MAYBE programming for kernel interfacing)

**Objective:**

1. to develop VGA Controller module for use on FPGA-based vision systems

**Scope:**

1. produce a working VGA controller module (HDL entry)

2. refer to [this](#)... maybe do higher resolution like SVGA? [more](#)...
3. must produce working interface for either linux kernel OR custom imaging logic
4. test and verify module on FPGA development board

## Development of Memristor Device Model for Spice Simulation

**NO TAKERS!** 

### Comment:

This project is purely software development but requires good background in low-level electronics (semiconductor physics).

### Keyword(s):

memristor, device model, spice

### Pre-requisite:

A in Computer Programming...? (This is NOT a “Hello, World!” programming project!), Good in Device Physics

### Objective:

1. to develop a spice simulation model for a memristor device

### Scope:

1. obtain open-source spice simulator code
2. learn how a model is created in spice
3. learn characteristics of a memristor
4. write a device model for memristor
5. build a simple test circuit
6. verify simulation results

## Extendable FPGA Development Board for Reconfigurable Computing Platform

**NO TAKERS!** 

### Comment:

This project is purely hardware development.

### Keyword(s):

FPGA, reconfigurable computing, development board

### Pre-requisite:

## Digital Electronics, Microprocessor System

### Objective:

1. to develop an extendable FPGA development board to be used as platform for reconfigurable computing research/applications

### Scope:

1. produce a working prototype for FPGA development board
2. design a working interface for future extensions
3. include/enable reconfigurable computing features in the design?

# Final Year Project 2011/2012

## General Requirements

Recommended development platform for all projects is Linux. Any distribution is acceptable, but [Slackware Linux](#) is recommended. A good alternative would be [Salix OS](#). I also recommend projects that involve source codes (e.g. HDL, C, C++) to use [source code management \(SCM\) or revision control software such as \[\[http://git-scm.com/ git\]\] or mercurial](#). I will be providing a wiki page for every project and I expect students under my supervision to keep a project journal within that wiki page. The template page will be given. To stress the importance of this journal, I will be making it part of your final evaluation marks. ===== General Comments ===== For each project title, I've included some keywords, objective(s) and scope. The keywords are there for you to do some background research before you jump into this - I don't want you to give-up on me (it's a waste of everybody's time) towards the end of the project. I put in objectives and scope to let you know what is expected out of this project. So, your results will be based on these expectations. I define an objective as the targeted final result. I usually set only ONE primary objective and optionally a secondary objective that will/should not affect your final results (though it usually gives you extra marks if completed) - too many objectives usually leads to unsatisfactory results. Meanwhile, scope visualizes how much work is expected - for example, one can say that the primary objective is to design a microprocessor. So, the scope should mention maybe something like you're expected to design a 4-bit microprocessor up to schematics level with instruction set that covers data transfer/processing and program control, complete with logic simulation results. ===== Project Titles ===== Development of an ARM-compatible Microprocessor Core for FPGA-based CPU/Controller ===== **Comment:** This is a multi-discipline project and requires knowledge in both hardware and software development. **RESERVED Keyword(s):** Microprocessor Core, ARM, FPGA, HDL Coding for Synthesis, Linux Kernel **Pre-requisite:** Microprocessor Systems, Embedded System Design... **Objective:** *still reviewing this (just a general description - mainly revolves around the idea of building a microprocessor core)* - to develop an FPGA-based general purpose CPU or controller board - (optional) to develop a mobile robot control/navigation systems using customized FPGA-based controller board **Scope:** - develop a complete microprocessor core on an FPGA (development or customized) board - design interface between the microprocessor core and the peripherals available on the FPGA board - design reliable testbench to verify the microprocessor core operations/interface - compile an operating system (e.g. Linux-based) for the board to create a general purpose CPU/controller board - test results for analysis and discussion/upgrades ===== GIT Web Interface (with Access Control) Using PHP ===== **Comment:** This project is purely software development. This is a 'support' development - 'end product' is to be

used by this group. **Keyword(s):** git, source code management (scm), php, web-interface, access control **Pre-requisite:** A in Computer Programming...? (This is NOT a "Hello, World!" programming project!) **Objective:** - to develop a web interface with access control for GIT repositories using PHP **Scope:** - can be based on [this](#), or write your own git backend engine - add access control, preferably using simple file DB - test on a live server ===== Development of Vision-based Driver Assistance




System (DAS) ===== **DID NOT DELIVER!** **Comment:** This can be a full hardware implementation (i.e. FPGA-based) or partly software (i.e. codes running on PC/controller). In any case, it requires a good understanding in basic image processing and artificial intelligence. **Keyword(s):** Vision systems, lane departure warning, obstacle and collision warning, traffic signs recognition, pedestrian monitoring, lane change assistance, blind spot monitoring **Pre-requisite:** Microprocessor Systems, Signal & Systems... **Objective:** - to develop a Driver Assistance System with at least 1 of the listed features **Scope:** - can be either software or hardware implementation (or even both) - a working prototype (at least 1 feature) at the end of the project - design reliable test (software simulations/in-the-lab execution) methods - test results for analysis and discussion/upgrades =====



Development of 8051 Development/Training Kit ===== **NO TAKERS!** **Comment:** This project also involves a combination of hardware and software development. **Keyword(s):** 8051 microcontroller, compilers & assemblers, software library, SDK, FPGA **Pre-requisite:** Microprocessor Systems, Embedded System Design... **Objective:** - to develop an 8051 microcontroller development and training kit **Scope:** - develop a basic 8051 soft-core to be implemented on FPGA - develop assembler/compiler for 8051 (optional: worst case scenario, use existing) - develop a software C library for various common devices (LCD, keypad, etc.) - design reliable testbench to verify the core operations/interface ===== Mapping (and Localization, maybe) for Mobile Robot Platform Using



Player/Stage ===== **NO TAKERS!** **Comment:** This project is purely software development but with embedded systems application. **Keyword(s):** mapping, localization, simultaneous-LAM  (SLAM), mobile robot, player/stage software **Pre-requisite:** A in Computer Programming...? (This is NOT a "Hello, World!" programming project!) **Objective:** - to develop a practical mapping and localization algorithm/application for mobile robot platform **Scope:** - using [this](#) - something like [this](#) - test on simulator platform only... ===== Final Year Project 2010/2011 ===== Recommended development platform for all projects is Linux. Any distribution is acceptable, but [Slackware Linux](#) is recommended. A good alternative would be [Salix OS](#). I also recommend projects that involve source codes (e.g. HDL, C, C) to use source code management (SCM) or revision control software such as [git](#) or [mercurial](#).

## ANNOUNCEMENTS

**201101211411** I've arranged for you guys to use MKR3 as a 'central' workplace. The official time that I've set is THU (1000-1500) and FRI (0800-1300). If you want, you can see En. Che Mohammad (Tech for that lab) to arrange for other times. I will be visiting you formally during that two days every week starting week 20110124-28. Please take note!

## Project Titles

### Implementation of an 8-bit/32-bit Microprocessor Core on FPGA

- can be RISC or CISC
- do analysis with picoblaze/microblaze as benchmark?

- implementation, testing, and improvise (if possible)
- should be expandable to micro-controller implementation

## Random Number Generator (RNG) on digital hardware (i.e. FPGA)

- based on [this page](#)
- implementation, testing, and improvise (if possible)
- testing is essential - maybe design an alternative test-suite to the one used

**Update** This title may have been canceled by FYP Panel

## Design and Implementation of an FPGA-based System

- board design based on selected FPGA (candidate: Spartan3E 500K gates)
- at least with RS232 interface, ethernet is nice to have
- obviously, need to be programmable
- external memory (e.g. SRAM) must be there
- for power requirements, [this](#) is a good reference
- it's nice to have [this](#) as well

## CAD Tool Development: Circuit Simulator

- must be on linux
- use open source parser or write own (can be a project on its own?)
- start with logic simulator, must be capable of timing simulation (timing model)
- alternatively: write a VHDL version of the [Icarus Verilog](#)?
- another branch: start from spice3f5 - get it to compile on linux & improvise

## CAD Tool Development: Automatic Place & Route

- must be on linux
- use open source parser or write own (can be a project on its own?)
- can be board-level or ic-level (select input netlist type and output format)

From:

<http://azman.unimap.edu.my/dokuwiki/> - Azman @UniMAP

Permanent link:

<http://azman.unimap.edu.my/dokuwiki/doku.php?id=archive:ekt444>

Last update: **2020/08/29 10:28**

