The EECS 3215 Design Project

One sentence overview

This is your chance to get hands-on "beyond Arduino" experience on a topic of your choice.

Overview

Each student is responsible for creating a hands-on project based on the LPC802 or LPC804 microcontroller or an FPGA of your choice. Students are responsible for costs and materials related to the project. We discussed this in Class 2.

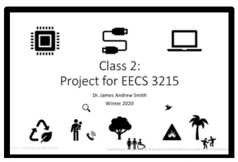


Figure 1 An overview of the project was given in Class 2. See the PDF on Moodle.

Grade weighting

The project is worth 20% of your final grade

Initial Submission (5%)

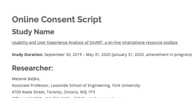
1. Written Abstract (1%)
2. System Diagram (4%)
   a. Project Pathway with Contingency (2%)
   b. Project Architecture (2%)
3. Due on Feb 28, 2020

Final Submission (15%)

1. Five minute video (7%)
2. Five page report (8%)
3. Due on April 6, 2020

Make sure to fill out the consent form for the study on making videos with your smartphone video:

1. https://tinyurl.com/qqxxszc
2. https://bit.ly/2SIjGv7



Online Consent Script

Study Name

Usability and User Experience Analysis of SmART, a on-line smartphone resource toolbox

Study Duration: September 30, 2019 – May 31, 2020 (January 31, 2020, amendment in progress)

Researcher:

Melanie Baljko,
Associate Professor, Lassonde School of Engineering, York University
4700 Keele Street, Toronto, Ontario, M3J 1P3
Office: LAS 2028, *416-736-2100 x* 33348, mb@eecs.yorku.ca

Project topic

The topic is up to you. It needs to be related to embedded systems. What you need to do is:

- Solve a problem with an Embedded System
- Design and implement the solution
  - Use a microcontroller (LPC80x) or FPGA
  - Integrate external components (like sensors, communication, motors and/or displays)
  - Program the micro or FPGA
  - Test the system
  - Demonstrate the system

Project technical scope

The project's technical scope is variable, just as the topics are.

The general "rule of thumb" is:

1. Two sensors (inputs)
2. Two actuators/displays (outputs)

This assumes that the sensors or actuators are relatively "basic." Three of each might make sense, depending on your design. If you were using a very complex sensor, then only one would be sufficient.

Consider the following:

- Your experience level dictates complexity
  - 1st timer? Simple off-the-shelf, no soldering
  - Old hat? Solder the board or try complex COTS (commercial, off the shelf)
- Use discrete components (individual chips, if you're comfortable)
  - Multiple discrete chips & support hardware
    - e.g. RS485 chip + power supply + support components
- Combine off-the-shelf boards… for example
  - e.g. 1 or 2 Arduino Shields
  - e.g. 1 or 2 Mikroelektronika Click Boards
- Integration of system
  - Breadboard is good
    - Pay attention to clean wiring (aesthetics and reliability)
  - Soldering is harder but better
- Packaging, Power & Display
  - Cardboard box is good
  - Wood, plastic or metal is better

Video work

Do the SAMBA project toolbox survey (https://bit.ly/2SIjGv7) and use the tools at the SAMBA project website to make a really good video.

Important dates

Part 1 due date: Friday, Feb 28 at 11:55pm via Moodle.
Part 2 due date: Monday, April 6 at 11:55pm via Moodle.

Suggested topics

Some of you have already come up with project ideas and have discussed them with me. If you are still struggling with coming up with a topic consider the following and a suggested list.

Your project should be thought of in the context of your final year capstone project.

- Automated door opener
- Toy trainer controller
- Bird feeder
- Solar powered calculator
- Clock display
- Tamagotchi
- Doorbell and remote chime
- Automated safe
- Simple "Roomba-like" cleaner
- Kinetic sculpture
- Musical instrument
- Weather station
- Home alarm
- Wireless telemetry or data logger (LoRa, ZigBee)
- Anything Bluetooth related (*generally hard!*)
- Background sound alarm
- Building temperature monitor (indoor vs. outdoor temperature)
- Window blind controller
- NFC-controlled inventory system
- … will add more suggestions to the Moodle page

> Dangerous projects are prohibited. That means that medical projects, anything to do with direct connections to AC power, anything with sharp objects or cutting surfaces are not permitted. There are plenty of other possible projects that are dangerous and we're not going to deal with those in this class. If you are thinking of doing something that might be considered dangerous, please come to speak with me.

Part 1: Initial Submission

In the first part you are to submit a plan for your project. It is at this stage that we verify your design and let you know whether you're on the right path for the final submission at the end of the semester. This consists of:

1. Written Abstract (1%)
2. System Diagram (4%)
    a. Project Pathway with Contingency (2%)
    b. Project Architecture (2%)

Together, it shouldn't take up more than about two pages of a document on "letter" sized paper. **The abstract** is one paragraph and should follow this model, complete with the bolded inline headings. Note that this description is for a very ambitious project. It's understood that your project will likely be less ambitious, and that's okay to still "meet expectations".

> **INTRODUCTION**: My embedded system is an automated dog feeder. **CONTEXT**: Dogs get hungry during the day. To prevent dogs rioting in the street and then taking over the planet, we will provide them with automated food dispensers. **TECHNICAL REQUIREMENTS**: The system has to cost less than $100, provide 100 ml of food three times in one day, weigh less than 5 kg, with food, and need a battery recharge only once a month. **SPECIFICATIONS**: [not included during Part 1] [***Proposed***] **COMPONENTS LIST:** 3 AAA batteries, a NiMH charger, my LPC802 board, a MikroE infrared sensor, an Arduino Motor Amplifier board, a solenoid and a custom-made plastic box to hold the food. **MY EXPERTISE LEVEL:** I have never built an embedded system before. **PROCEDURE**: I will purchase my parts directly from Digikey and will develop the system on a combination of custom soldered circuit board and a MikroE Click Shield. I will program the system in C++14. **CONTINGENCY**: In the event of difficulty I will develop code in C, will use a breadboard for the circuit and/or will only detect the dog using the IR sensor without implementing a solenoid-based food dispenser.

Commented [JAS1]: Short and sweet. What does it do?

Commented [JAS2]: Social, environment or political context. Think about it. Are you just making a device because "it's cool" or can this be used by people for a good reason? If it's just "cool" then state it here. Want help with contextualization? Come and talk.

Commented [JAS3]: REQUIREMENTS are the informal precursor to the SPECIFICATIONS. We need a little more detail than the intro. This will complement the Architecture drawing. You'll be more formal about the "specs" in the Final Submission.

Commented [JAS4]: Usually you would provide a detailed specfciations list. The Requirements and Architecture diagram are sufficient for this, your first attempt at embedded design.

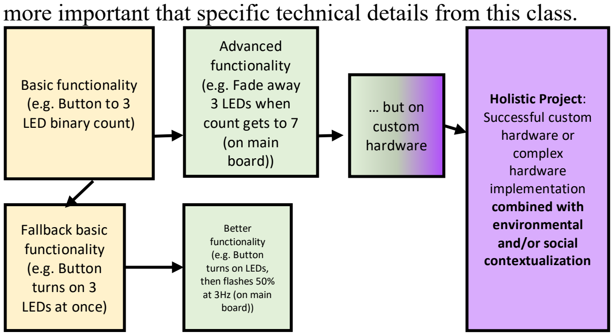
Commented [JAS5]: This may change, but you should have a solid plan.

Commented [JAS6]: Be honest.

Commented [JAS7]: Backup plans are key.

Project Pathway with Contingency

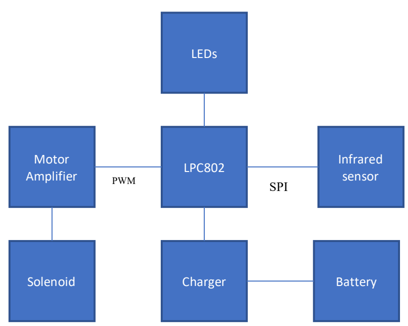
It's important to have a plan for implementing your project. Sometimes, for many reasons, your plan doesn't go as you had wanted. Plan ahead so that you can make the decision to fall back on a "contingency" version of your project. Below is an example of that. Your contingency project won't get you the best grades, but you will have learned some valuable lessons that are, arguable, more important that specific technical details from this class.



The Project architecture.

The Wolf book has good examples of this. It's basically a block diagram showing the major parts and how they connect together. Powerpoint, Dia, Inkscape and Visio are all good tools for this.

This is a block diagram showing the different components and how they connect to each other. Less detailed than a schematic diagram. For instance, while in a schematic you might show four different wires for SPI serial communication between the LPC802 and a sensor and you would show to which pins these wires connect, in the architecture illustration you just show one line between the two components and call that line "SPI."

	*[Not for Part 1. You'll work on this for the final submission. It should look similar to the hardware architecture but describe the software components (data objects, functions, ISRs, etc.)]*
Hardware Architecture	Software Architecture

Note that each project should have, at its core, a way of keeping track of elapsed time since power-up in 10 ms increments using SysTick.

Grading for Part 1

Rubric for Part 1	CLO 3: **Design, implement and interface with standard and custom peripherals** (*GAI: Conceive design solutions to solve the defined problem*) 1. **Does not design solutions to solve defined problem.** 2. **Designs incomplete solutions.** 3. **Solutions complete, but lack in elegance/innovation/creativity/professionalism.** 4. **Conceives elegant/innovative/creative/professional standard solutions to solve the defined problem**

Detailed Marking Guide for Part 1

Each component in Part 1 should tell part of the "conceive design solutions" story that illustrates how you are "designing, implementing and interfacing with peripheral devices" (external to the microcontroller or FPGA).

Below is the marking guide that the TA will use when evaluating your submission.

Written Abstract	Grade	CLO 3: **Design, implement and interface with standard and custom peripherals** (*GAI: Conceive design solutions to solve the defined problem*)	Note
	/4	**1. Does not design solutions to solve defined problem.** **2. Designs incomplete solutions.** 3. **Solutions complete, but lack in elegance/innovation/creativity/professionalism.** **4. Conceives elegant/innovative/creative/professional standard solutions to solve the defined problem**	
Project Pathway (w/ Contingency)	Grade	CLO 3: **Design, implement and interface with standard and custom peripherals** (*GAI: Conceive design solutions to solve the defined problem*)	Note
	/4	**1. Does not design solutions to solve defined problem.** **2. Designs incomplete solutions.** 3. **Solutions complete, but lack in elegance/innovation/creativity/professionalism.** **4. Conceives elegant/innovative/creative/professional standard solutions to solve the defined problem**	
Project Architecture	Grade	CLO 3: **Design, implement and interface with standard and custom peripherals** (*GAI: Conceive design solutions to solve the defined problem*)	Note
	/4	**1. Does not design solutions to solve defined problem.** **2. Designs incomplete solutions.** 3. **Solutions complete, but lack in elegance/innovation/creativity/professionalism.** **4. Conceives elegant/innovative/creative/professional standard solutions to solve the defined problem**	

Want to exceed expectations?

- Go <u>beyond</u> an excellent technical widget.
- Context and Impact of your widget
  - Social
  - Political
  - Environmental
- Researched
  - Citations to existing technologies, issues
  - Mix "popular" & "academic" sources (scholar.google.com; library.yorku.ca)
- Alternative
  - Excellent Technical Widget
  - Wikipedia entry in Embedded domain
    - Translation
    - Person profile (new; non-traditional)
    - Technical entry that addresses equity, diversity, inclusion issue

Part 2: Final Submission

The final submission is due in April. You are to submit both a report and a video. Both the report and the video will be evaluated according to the course learning outcomes. Each is supposed to tell the story of your design work. The video is to be about five minutes long. The report is to be approximately five pages long. Read on for details.

The Report

Make sure to use the following headings in your report.

INTRODUCTION
- A short description (three to five sentences)

CONTEXT:
- Describe "what" and "why"

TECHNICAL REQUIREMENTS
- Relatively informal things that the system should do. As per the Wolf book.

TECHNICAL SPECIFICATIONS:
- As per the Wolf book. These are more detailed that the requirements. A drawing (schematic) can be included here.

COMPONENTS LIST: [*as you built the device*]
- What was in your system? Include two architecture drawings: one for hardware and one for software.
- A photo of the system is appropriate here.

PROCEDURE:
- Describe the process that you used in creating your project.

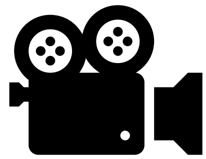
TEST:
- how did you test that the system worked?
- Got graphs showing the results of tests? (power, etc.)

CONTINGENCY
- Did you have to engage your contingency? If so, why? Reflect on this. What would you do differently next time. Looking ahead to ENG 4000 are there any lessons you learned that you would like to apply?
- Include your project workflow from Part 1, including any changes regarding the continency.

ADDITIONAL MATERIAL
- Especially, anything to do with making your project "holistic". Tie in context of the project and discuss any social, political or environmental aspects about your project that you believe makes it a candidate for "exceeding expectations."

CONCLUSION
- Wrap up in a few sentences.

The Video



The video is to be about five minutes in length. The video format and approach are more free form than the written report. Keep in mind that the assessment is not free form, though. It is evaluated the same way, using the rubrics as the written report. During your video you need to not only showcase your project, but you must also address aspects of the following.

Fill out the consent form and complete the follow-up survey on using your smart phone to create videos for class: https://bit.ly/2SIjGv7

Review this video for general skills **about video creation** for student projects:

- https://www.youtube.com/watch?v=pzDbPOg1rxs

You should be creating a **narrative video**, with voice over or text subtitles. What does a narrative video look like? Have a look at this example:

- http://samba-toolkit.eecs.yorku.ca/knowledgebase/what-does-a-narrative-video-look-like-when-submitted/
- Alternate link: https://www.youtube.com/watch?v=KjycUDQHcnE

When planning your video, please review **best practices** to ensure that you are creating the best possible video to showcase your project while also respecting the **privacy**, security and safety of others. Review

1. **Ethics** best practices: http://samba-toolkit.eecs.yorku.ca/video-media-release-best-practices/
2. Ethics **decision trees**: http://samba-toolkit.eecs.yorku.ca/media-release-tool/

Also, you want to demonstrate the **best technical featu**res of your project. Have a look at this example of a Nursing student illustrating technical skills. What technical skill from your embedded systems project can you illustrate in your video?

1. http://samba-toolkit.eecs.yorku.ca/knowledgebase/what-does-a-technical-student-skill-look-like/

Detailed Marking Guide for Final Submission (Part 2)

We will be evaluating the following four learning outcomes via their general GAI attribute descriptions. **Both** the **Report** and the **Video** will be assessed using these rubrics.

Report (CLO 1)	Grade	CLO 1: **Select and utilize appropriate parallel, serial and analog interfaces** (*GAI: Use specialized engineering knowledge of design specific components, systems or processes to solve engineering problems*)	Note
	Report /4 Video: /4	1. Does not to use specialized knowledge needed to solve the engineering problem 2. Uses some specialized knowledge of design specific components, systems or processes to solve the engineering problem 3. **Uses appropriate/relevant specialized knowledge of design specific components, systems or processes resulting in a reasonable solution** 4. Sophisticated use of specialized engineering knowledge of design specific components, systems or processes to solve engineering problems	
Report & Video (CLO 2)	Grade	CLO 2: **Design embedded software and hardware systems to address problems in important application domains under tight constraints** (GAI: Demonstrate skills in computer programming, data analysis and graphical visualization)	Note
	Report /4 Video: /4	**1. Does not demonstrate skills in computer programming, data analysis and graphical visualization** **2. Demonstrates marginal skills in computer programming, data analysis or graphical visualization** **3. Demonstrates competency in computer programming, data analysis and graphical visualization** **4. Demonstrates superior skills in computer programming, data analysis and graphical visualization**	
Report & Video (CLO 3)	Grade	CLO 3: **Design, implement and interface with standard and custom peripherals** (*GAI: Conceive design solutions to solve the defined problem*)	Note
	Report /4 Video: /4	1. Does not design solutions to solve defined problem. 2. Designs incomplete solutions. 3. **Solutions complete, but lack in elegance/innovation/creativity/professionalism.** 4. Conceives elegant/innovative/creative/professional standard solutions to solve the defined problem	
Report & Video (CLO 4)	Grade	CLO 4: **Prototype embedded systems using microcontrollers or field programmable gate arrays (FPGAs)** (ignore GAI 1; GAI 2: Decompose complex systems into smaller, more manageable sub-systems.	Note
	Report /4 Video: /4	**1. Does not decompose complex systems into smaller, more manageable sub-systems; or proposed sub-systems are incomplete or illogical** **2. Able to decompose complex systems into smaller, more manageable sub-systems but missing one or two sub-systems (incomplete)** **3. Able to decompose complex systems into smaller, more manageable but suboptimal sub-systems** **4. Able to decomposes complex systems into an optimal set of smaller more manageable sub-systems**	

More on the Rubrics & Marking Guides

In many Lassonde courses, like ENG 4000, we use qualitative criteria to assess students. Both students and faculty have noted that it gives a very different flavour to assessment. Most students will end up in the meeting expectations category, which maps to a "B" to "B+" range of traditional grades.

Rubric Numeric Score	Rubric Description	York Description	York Letter Grade
4	Exceeding Expectations	Exceptional	A+
3.5		Excellent	A
3	Meeting Expectations	Very Good	B+
		Good	B
		Competent	C+
2.5		Fairly Competent	C
		Passing	D+
2	Marginally Meeting Expectations	Barely Passing	D
1.5		Marginally Failing	E
1	Below Expectations	Failing	F

While you may aspire to a perfect ("100%") grade, the reality is that very few actually do on a properly designed learning activity. Yes, there will be some of you who will both aspire to and achieve an exceptional result in your project and, so, your project may be categorized as "exceeding expectations". It's not that others have done anything "wrong" but, rather, they have met the targeted learning; that is, they have "met expectations" of learning, which corresponds, when mapped to a traditional score, to B or B+.

How will this play out in the project? To at least "meet expectations" you need to do so in all of the assessment components, both for the submission after reading week and for the submission at the end of the semester. This, typically, will mean that your project has also met all of its technical goals and that you didn't have to fall back on your contingency plan.

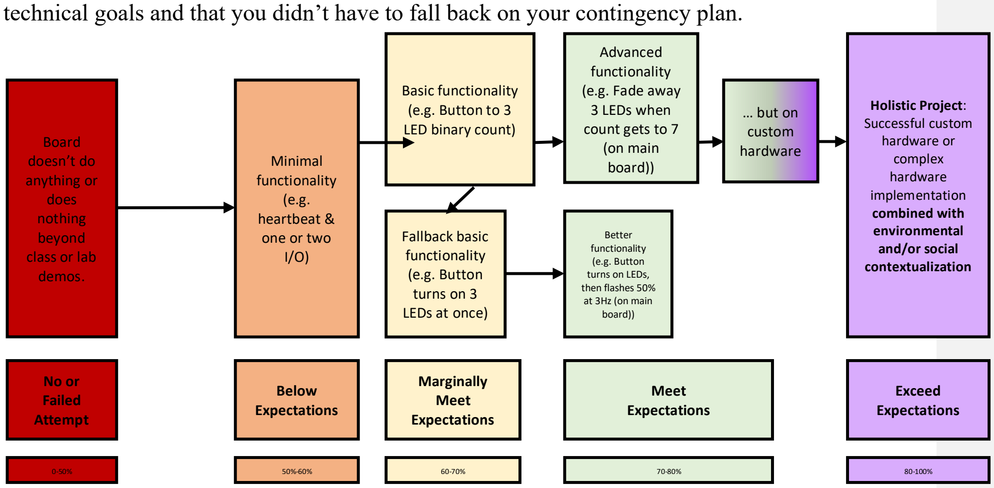
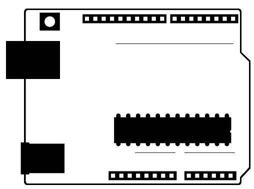


Figure 2 Overview of rubric assessment in context of project development. Note the built-in contingency pathway.

Why not just use an Arduino for the Project?[1]

Arduinos are awesome. How awesome? Super awesome. But not EECS 3215 awesome.

The Arduino system and the associated "Maker Movement" changed everything about embedded systems. We teach embedded systems differently today because of Arduino. Development boards now come with stackable headers that are "shield" compatible because of Arduino, in spite of an original design flaw (!) in the UNO layout. C compiler toolchains are arguably free today because of the Arduino sketch IDE. C++ is also arguably available for embedded devices because of Arduino (the Arduino's ATMEGA328 is the only mainstream 8-bit device that has C++ support).



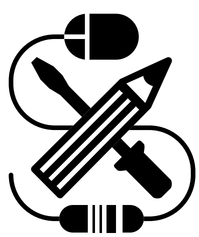
Coupled with manufacturing facilities in China, prices for development boards and debugger tools have plummeted to near zero. Embedded systems are better because of Arduino.

> My goal, as an engineering prof, is to give you the skills to choose whichever tool makes the most sense for the projects you want to do.
>
> When it's a simple, quick and dirty and you're not going into production, then choose Arduino. I do this all the time.
>
> But Arduinos are limited. Other microcontroller solutions don't have their limitations. The limits include:
>
> - Libraries that only work for select applications
> - Limited hardware set that might not meet your project specifications
>
> EECS 3215 is an opportunity for you to explore general skills that will help you meet any project specifications you encounter in the future.
>
> -James

But Arduino is a double-edged sword. The ease of use of Arduinos has meant that non-engineers now make devices that previously used to be the exclusive domain of engineers or computer scientists. That means that there is now] you're not the only game in town. So, to compete or to distinguish yourself you need to "up your game."

Upping your Embedded Systems Game

In short, while everyone can use Arduino, only a few people can write the underlying libraries for Arduino or develop hardware boards that are compatible with Arduino. That's where you fit in. The skills we learn in courses like EECS 3215 are to provide you with the insight to be able to do things that other people find hard.



Others can <u>use</u> Arduino, you can <u>build</u> <u>for</u> Arduino.

[1] Arduino image: noun_Microcontroller_328364.svg (The Noun Project) ; Maker image : noun_Maker_18157.svg