UNIVERSITY OF BRITISH COLUMBIA
MECH 423 – Mechatronic Product Design 2022W
Final Project – Handout #1/2
Overview, Logistics, Proposal, and Potential Projects

Overview

The MECH 423 final project aims to motivate students through the process to design and build a mechatronic product of their own invention. Students work in teams of two to formulate a project objective, divide this objective into distinct functional requirements (FRs), and outline a plan for their work in a proposal. Each team will then design, build, and test modules to satisfy their FRs before assembling and testing the overall system. Finally, each team will create a video of their project, demo their project, and write a final report.

Students are free to develop any device that appeals them, as long as the development of the device could be divided into a set of reasonable FRs. A major exception is that projects aimed to develop weapons, even benign weapons, are not permitted. Each project is required to have at least five FRs. Standards for what constitute a valid FR for MECH 423 will be discussed in class. Students may choose to invent a new product or to develop an existing product using their own electrical and mechanical components.

Students are encouraged to be creative and ambitious in their project proposal, but they are also encouraged to be prudent in choosing a project compatible with their technical capabilities and time availability. Specifically, students should aim to find projects that maximize the impact for their efforts, which is to say that students should aim to use minimum engineering effort to achieve maximum effect. For example, a project to create a remote-controlled toy car is unlikely to have this property since it is very difficult for students to make something better or more interesting than a well-designed, mass-manufactured toy car. Importantly, students need to produce a working prototype that is robust and can perform consistently when used multiple times. Therefore, it is important to ensure the FRs are realistic given your capabilities and resources. An important part of the exercise is for students to estimate the difficulties associated engineering development tasks in mechatronic devices.

Microprocessors

We require students to use the MSP430 microprocessor as the primary microprocessor for their projects. The reason for this requirement is that students often underestimate challenges in working with different microprocessors. Since MECH 423 is taught using the MSP430 microprocessors, challenges that arise are typically more easily resolved than other microprocessors. Use of other microprocessors to provide secondary and tertiary roles may be possible, but students must first obtain approval from the TAs.

Materials and Supplies

The TAs will set up class orders each week. Student teams are expected to acquire some basic supplies from available sources, such as fasteners from ECE stores and Mech shop, materials from surplus, and components from e-waste. MECH 423 will provide each student team with up to $100 in materials purchased from Digikey, McMaster, and Amazon. Student teams planning to use specialized components in their project are expected to purchase the components themselves.

A parts library is available for student teams to borrow components for their project (descriptions of available components are in a separate document). Student teams must first obtain permission from the TAs to borrow components. Borrowed components must be returned at the end of the semester in usable condition.

The final projects, with the exception of components from the lab kit and parts library, are for the students to keep after the completion of MECH 423.

Evaluation and Due Dates

Project component	**Grade**	**Due date**
Chat with instructor/TAs	0%	Oct. 31
Project proposal due	10%	Nov. 7
Project demo including: • Completion of each FR • Overall product based on functionality, difficulty, creativity, and sensibility	60%	Dec. 9 5-10 pm KAIS 2020
Video (content, quality, and length)	15%	Dec. 11
Final report	15%	Dec. 11

Deliverables

Please provide the following materials with the final project report. A Google drive folder will be provided to facilitate material transfer.

- Code for labs 1, 2, and 3 (for accreditation purposes)
- Project proposal
- Project report
- Project design files
- Project presentation slides
- Project demonstration video (must be <100 Mb)

Video

Your video should demonstrate key features of your working prototype. Your video should be short and to the point. The total length should be no more than 60 seconds and the total file size should be <100 Mb. Your video should be either narrated or contain text or slides that explain the hardware to the viewer. Most of the video editing functions you'll need can be found on free programs like Windows Movie Maker or iMovie. Your video may be used to promote future iterations of MECH 423, as well as the Department of Mechanical Engineering. Please let us know if you prefer not to have your video made publicly available.

Project Proposal

The proposal describes the objectives and motivations of the project, divides the project into distinct FRs, and presents the detailed design and work plan for each FR. Use the following outline when drafting the project proposal. Detailed descriptions of each proposal section are presented subsequently.

1. Objectives
2. Rationale
3. List of Functional Requirements
4. Functional Requirement #1
    4.1. Approach and Design
    4.2. Inputs and Outputs
    4.3. Parameters

Detailed Description of Proposal

1. Objectives

Describe the overall goal and vision of your project. Describe what you will design and build. Describe what the final product will look like.

2. Rationale

What is conceptually interesting about your project? Why is it worth your time to do it? How is it different from similar projects at UBC and elsewhere?

3. List of Functional Requirements

Separate your overall system into distinct FRs. Briefly describe each FR and the hardware and/or software that will be developed to address each FR. Describe how signals are transferred and manipulated within your system. Use graphical illustrations if appropriate. Create a table containing all your FRs. Assign a percent effort value for each. Identify which team member will be responsible for a specific FR. For two person teams, only one FR can be shared between team members.

Functional Requirements	**% Effort**	**Responsible Person**
FR#1: Sensor interface circuit	20	Calvin
FR#2: MSP430 code to measure angular speed from motor encoder	20	Calvin
FR#3: MSP430 code to control LCD screen	25	Hobbes
FR#4: Motor driver and interface electronics	15	Hobbes
FR#5: User interface programmed using C#	20	Calvin and Hobbes

4. Functional Requirement #1

For each functional requirement in your system create subsections that address the following.

4.1 Approach and Design

Describe the objective of the FR. Describe the hardware and/or software that will be developed to address the FR. Present preliminary design material such as circuit diagrams, algorithms and pseudo-code, mechanical sketches, solid models, block diagrams, and screen-shots from proposed user-interface software.

4.2 Inputs and Outputs

Describe the inputs and outputs of the hardware and/or software module developed to address the FR. Examples include voltages representing physical parameters, serial data streams, and user-interface outputs. Present details such as the transfer function and range for analog signals, the format of serial message packets, and text and graphics presented to the user.

4.3 Parameters

Identify the parameters of the hardware and/or software module developed for this FR. (*i.e.* what are the knobs that need to be adjusted on this module?) Examples include sensor bias voltage, values of key circuit components, motor operating speed, dimensions of mechanical elements. Describe how these parameters affect the operation of the module. Discuss how you will optimize these parameters.

4.4 Development Plan

Separate the development process into incremental steps. Present these steps as a list.

4.5 Test Plan

For each step in the development plan, describe a simple test that will be performed to verify hardware and/or software functionality.

5-8. Functional Requirement #2-5

9. Most Critical Module

Identify the most critical module of your project. Devise preliminary experiments to verify the feasibility of your MCM.

10. Risks and Countermeasures

Identify potential conditions that could undermine the success of your project. Discuss countermeasures when these unfavourable conditions arise.

11. Learning Objectives

Describe what you hope to learn by doing this project.

Project Ideas

The projects listed below are designed to be starting points for students to develop a proposal. Feel free to modify any parts for your own proposal, or preferably, create a proposal from scratch.

Motion and Light Activated Bicycle Light

LED lights are a nice bike accessory in case you had to do some nighttime riding. However, they have to be turned on, which slows you down and can run down the battery if you forget to turn them off. Design a bicycle LED that only turns on while the bike is in motion and in low-light environments. Keep manufacturing costs to a minimum by using a low-cost microprocessor, light sensor, and devising your own low-cost motion sensor. Develop strategies to keep parasitic power consumption to a minimum. Estimate the battery life of your device. Student teams should produce a polished final product.

Remote Weather Station

Develop electronics and software for a remote weather station to measure the temperature and wind speed at the top of Whistler. Use the MCP9700-E/TO-ND sensor to measure temperature. Build a thermal anemometer using precision resistors as heating elements. Calibrate wind speed and direction in the wind tunnel. Report the temperature and speed to a C# user-interface. Use the C# program to send periodic emails or tweets of current weather conditions with time and date stamp.

Building Ventilation Controller

Ventilation is a major source of energy consumption in public buildings. Develop a building ventilation controller that changes the amount of ventilation depending on the level of activity in the building. Use a microphone to estimate the activity level in the building. Use the motor as your ventilation fan. Use the building activity level to control the motor speed. Create a user interface using C# or a LCD screen to show the building activity level and the fan speed. Allow the user to vary the relationship between activity level and fan speed.

Home-made Pen Plotter

Bubblejet printers are an incredibly cheap way to obtain a precision 2-axis stage. Find an old bubble jet printer. Open the printer and remove the cable to the stepper motors from the control board while keeping the stage intact. Build a motor controller for the two stepper motors. Use MSP430 to generate control sequences for moving the motors forwards and backwards. Use C# to send commands like “forward 100 steps” or “left 20 steps”. Replace the bubblejet cartridge with a pen or pencil. Demonstrate plotting simple figures on a sheet of paper. Other modes of artistic patterning include wood-burning and foam cutting.

New Interfaces for Musical Expression

An often heard complaint at electronic music concerts is that the artists are simply “playing” their laptop. Develop new interfaces for musical expression as a means to connect physical gestures with music and enable new ways to generate and improvise music. Develop a platform for generating music using physical gestures. Use Ableton Live to create electronica melodies and then use signals from the accelerometer, motor/encoder, and push buttons to trigger MIDI inputs on your computer. Use the LCD screen and soft-keys to add trigger dynamic content (changing with song content).

Colour-changing kinetic sculpture (Heavy mechanical component)

Kinetic sculptures are moving art displays. Create a kinetic sculpture using one or more motors. Change the sculpture depending on the ambient light-level, temperature, and noise level in the room picked up a microphone. Add LEDs to the sculpture and change their colour and brightness output depending on the state of the sculpture. The sculpture could be a designed as a permanent system or perhaps as a flashing toy that is tossed around (i.e. at a party or a concert).

Lie Detector

Create a lie detector that monitors various physiological signals when a person is asked a series of questions. Potential measurement parameters include heart rate, motion (fidgeting), face and hand temperature, and skin response (sweating). Potential ways to measure heart rate include ECG sensor, pulse oximetry, and microphone. The ECG sensor is potentially a difficult circuit to build. A shortcut is to purchase an ECG sensor and extract the ECG waveform by hacking a wire to the amplified output. Motion could be measured using accelerometers. Skin response could be measured using galvanic skin response sensors or home-brew versions that measure skin conductivity.

Simon Says (Heavy mechanical component)

The game "Simon Says" game is a pattern matching game, where an increasingly difficult pattern is displayed and the player of the game must repeat that pattern in order to go to the next level (more elements are added to the end of the pattern each level). Create this game with a new twist: use visual indicators, sensors (buttons or touch sensors), and haptic feedback to enhance the player's experience.

Knock Pattern Door Lock

Forgetting your keys and calling up a roommate to let you in is always frustrating, or trying to find a place to hide keys that's not as obvious as under the door mat. Inspired by the "What's the secret knock?" days of our childhood, develop a door lock that will activate when a specific knocking pattern is detected. Commercially available solenoid door locks can be used as a starting point. Use a contact microphone to pick up the knock pattern from a door. The knock pattern could have a temporal component (time between knocks), as well as a spatial component (location on door). Make sure this system cannot be easily fool by random knocking.

Whack-A-Mole Game

Re-make the classic game or the following new version: There are a number of tiles, any of which could be indicated (for example with a light). The player then has a certain amount of time to activate that tile using a sensor of your choice, and then another tile is activated. This repeats, with the time between tile activation decreasing until the player makes a mistake and the game is over. Create a version of this game, and optionally keep track of high scores through a Windows interface.

Additional Project Ideas

- Persistence of vision display
- Hack an existing display device to display your own messages
- Energy harvester
- Run-away alarm clock
- Animatronic dancing robot
- Self-balancing machine
- Ultrasonic wood stud finder
- Capacitive metal stud finder
- Home-away program
- Office temperature controller controlled using an Internet calendar (i.e. gmail calendar)
- Labyrinth-playing robot (start with a simple version!)
- Low-cost electrode-discharge machining tool
- Home-brew laser scanner display
- Automated pet food dispenser
- Location based games using a GPS chip
- Puzzle alarm clock (must solve the puzzle to disarm the alarm)
- Unuseless machine (electromechanical Rube Goldberg machine)
- Escape room
- Maze solving robot
- 3D block builder (like a 3D printer)
- Smartphone app to control hardware via Bluetooth
- Part picker/sorter

- Sunlight finding planter
- AR video game with motion sensing control