Developer's Guide To Pacemaker Development

SFWRENG/MECHTRON 3K04
McMaster University

Introduction to the Project

Fall 2025

Introduction

Welcome to SFWRENG/MECHTRON 3K04. For the course project, you will get to develop a system that operates a cardiac pacemaker! We would like to give you some helpful information on what to expect and how to complete the course project successfully.

The project will be conducted in teams of 5 to 6. All team members must be in the same lab section. During your first lab, if you do not have a full team, your TAs will assist you with finding a match with other students who are also looking to build a full team. Labs will begin in the second week of the semester and will be held every week (with the exception of the reading week and the last week of classes). At the start of the first few labs, your TAs will lead you through a brief tutorial about a concept related to the project. You will then have designated time to consult your TA, or continue work on the project with your group for the rest of the lab session.

You will also receive two hardware kits per team that you can use to develop and verify your pacemaker system. Your TAs will keep track of which kits your team will be responsible of. You will return them the last week of classes.

The course project is divided into two projects. You will be required to submit your code and documentation by the deliverable due date, and to demo your deliverable during your lab section during the week. The project deliverable deadlines have been shared during the first week of classes during lecture and you can find them in the course calendar in Avenue.

The goal of the project is to demonstrate your ability to apply engineering principles with the software development process. To do well on the project, keep in mind the software design principles while designing and implementing your software. Make sure your software is maintainable, reliable, robust and modular. The software must meet all requirements specified. It would be a good idea to build good test cases ahead of time and to test your software incrementally. Secondly when developing the user interface, keep in mind the safety and security aspects of the pacemaker. A good user interface also maximizes usability and user experience, and is intuitive, leaving no confusion about what the device is doing. In addition to designing, implementing and verifying the software, a major part of the course project is to read and write technical documentation effectively. No system is ever complete without technical documentation specifying the design, and guiding current and future development efforts. When writing your documentation, make sure it is informative, clear, concise, complete and correct. Also highlight design decisions made my your team. Do not leave the documentation to the last minute!

We recommend that you begin planning out your project ahead of time. You will have a fair amount of ground in order to cover to complete the project well and you will certainly need to have all hands on deck! Start by reviewing the project requirements and background material on the cardiac conduction system. As you gain a better understanding of the project, break down your work into as many granular tasks as you see fit and estimate how much time your team would need to complete each part of the project. Consider making a rough timeline of your objectives as well. As is the case with any successful project, it is imperative that you stay organized, and track your progress well so you can meet your deadlines ahead of time. It may also be a good idea to use a version control tool, such as git, to help you keep track of your software changes among your teammates.

If you find that getting started on the project seems overwhelming or daunting at first, don't despair. You will get the hang of it. This guide is designed to take you step by step through the process of learning the concepts you need for the project. And of course, you will have lots of support from your TAs along the way. Do not hesitate to ask us if you have any questions about the project.

We hope that the course project will prove to be a valuable learning experience for you and that you find it interesting.

— 3K04 Teaching Team

About This Guide

This guide is made up of 7 tutorials. Each tutorial consists of a summary section, a background section and a tutorial section. The summary section outlines the topics covered, and references suggested or relevant reading materials. The background section describes background material and motivation for the concepts discussed. The tutorial section outlines steps you can follow to implement the concepts.

- **Tutorial 1.1:** Pacemaker Project Overview
- **Tutorial 1.2:** Getting Started with the MATLAB Simulink Environment
- **Tutorial 1.3:** Hello World
- **Tutorial 1.4:** Getting Started with the Pacemaker Development Platform
- **Tutorial 2:** Pacing, Sensing and Timing Cycles
- **Tutorial 3:** Serial Communication
- **Tutorial 4:** Rate Adaptive Pacing

Tutorial 1.1 introduces the project and the pacemaker system.

Tutorial 1.2 introduces the environment you will be using to implement the pacemaker software.

“Hello World” is a time-honoured tradition in computer programming — a simple exercise that gets you started when learning something new. **Tutorial 1.3** aims to get you started on Simulink programming by building a basic Simulink model.

Tutorial 1.4 introduces the Pacemaker Development Platform — the platform you will be using to develop and perform tests on the pacemaker.

Tutorial 2 to **Tutorial 4** cover pacemaker functional concepts to facilitate building models for the pacemaker system that implement pacing controls, sensing logic, temporal logic, serial communication and rate adaptive pacing.

Lab Schedule

We will be posting **Tutorials 1.1, 1.2, 1.3, 1.4 and 2** with the **Deliverable 1** handout as they cover knowledge you need to work on it. The remaining tutorials will be posted with the **Deliverable 2** handout. Your team may decide to work on the multiple tutorials outside the lab, however, the TAs will be summarizing and walking through the content during specific lab sections. You will also have time to work on the activities and ask questions.

- **Week of Sep. 8th:** First week of labs!
  - Form your teams and inform your TAs
  - Work on **Tutorial 1.1** and **Tutorial 1.2**
- **Week of Sep. 15th:**
  - Complete and sign your team contracts and upload them to Avenue
  - Receive your hardware kits, verify every item is in it (as outlined in the **Tutorial 1.4** handout).
  - Work on **Tutorial 1.3** and **Tutorial 1.4**
- **Week of Sep. 22th:** Getting help from the TAs on the project.
- **Week of Sep. 29th:** Getting help from the TAs on the project.
- **Week of Oct. 7th:**
  - **Deliverable 2** will be posted.
  - Work on **Tutorial 2**.
- **Week of Oct. 20th:** Getting help from the TAs on the project.
- **Week of Oct. 27th: Deliverable 1** Demo week!
- **Week of Nov. 3rd:** Work on **Tutorial 3** and **Tutorial 4**.
- **Week of Nov. 10th:** Getting help from the TAs on the project.
- **Week of Nov. 17th:** Getting help from the TAs on the project.
- **Week of Nov. 24th: Deliverable 2** Demo week!

This schedule covers what you must work on during the lab. While there isn't a particular timeline for your project (other than the deadlines), it is your responsibility to make constant progress on it, and use the lab as a set time to work as a team and get help from the TAs.