Embedded Medical Devices

BME554L (Spring 2026)

Table of contents

Personnel

Instructor

Dr. Mark Palmeri (`mlp6`)

- Email: mark.palmeri@duke.edu
- Ed Discussion (private message, TAs included)
- Office Hours: Sign Up

Teaching Assistants

- Sophie Pelton (`smp147`)

For TA office hours (in the lab), please see the Canvas calendar.

> **Tip**
>
> Questions that can be answered by Dr. Palmeri or a teaching assistant should be posted on Ed Discussion.

Course Times & Locations

Lecture

Monday & Wednesday, 08:30-09:45, Hudson Hall 212 (Panopto recorded)

Labs

Location: CIEMAS B209

There are no formally scheduled lab sections, but you will need to use the lab equipment to perform testing of your devices. Your TAs will announce times when they will hold lab hours.

> **Important**
>
> Please review the lab policies before using the lab for the first time this semester.

> **Warning**
>
> No food or drink is allowed in the lab! Failure to adhere to this policy will have consequences on your lab grades.

Course Objectives

This course will give students experience with the design, function and deployment of embedded medical devices. Students will have hands on experience with electronic hardware and firmware (software) development, along with gaining experience with biosignal transduction into circuits.

Upon completion of this course, students should be able to:

- Version control software / firmware development using `git`.
- Use an Integrated Development Environment (IDE) for firmware development.
- Describe hardware using a Devicetree hierarchical data structure.
- Develop firmware using Zephyr as a bare-metal super-loop and a Realtime Operating System (RTOS)
- Implement state machines and generate state diagrams using the Unified Modeling Language (UML).
- Utilize callbacks / interrupt service routines for realtime event detection and response.
- Utilize threads and work queues.
- Develop firmware to control common peripherals, including GPIO, ADC and PWM.
- Use different serial communication protocols, including UART, I2C, SPI, and BLE.
- Utilize firmware logging at different levels.
- Test firmware implementation on the `nRF52833DK` using electronic bench equipment and generate technical reports with data analysis for device verification.
- Develop firmware with workflows that adhere to relevant industry and safety standards (e.g., UL, IEC60601, IEC62304) for FDA 510k clearance.

Prerequisites

Mandatory

- EGR105L or equivalent experience [git, Python]
- Intrumentation (BME354L) / Mechatronics
- Signals & Systems (BME271/671 or equivalent experience [filtering, FFTs])

Learning Management System

We will be using Canvas as the learning management system for this course. Most resources will be linked to the course website. All grades will be posted via Canvas/Gradescope.

Duke's GitLab server will be used for most course lab exercises, and code-related questions will be submitted to Dr. Palmeri / TAs using GitLab Issues.

Ed Discussion will be used for general course questions and discussion.

Class Schedule

This class is organized in a sequence of modules. Specific details surrounding dates for assignments associated with each module will be posted to Canvas/Gradescope and linked below.

This course uses a version of Mastery Learning, where "mastery" of a given module is necessary to progress onto the subsequent module. Quizzes are used to evaluate "knowledge"; lab exercises are used to demonstrate application of skills. In this course, assignments of later modules depends on the successful completion of earlier modules.

- Zephyr & Nordic SoC Overview
- Event-Driven State Machines
- Version Control with Git
- C Programming Overview
- Devicetree, GPIO, ISR, Callbacks
- Timers & Work Queues
- Threads & Kernel Events
- State Machine Framework
- Analog-to-Digital Conversion (ADC)
- Pulse-Width Modulation (PWM)
- Serial Communication Protocols: UART, I2C, SPI
- Bluetooth Low Energy (BLE)

Attendance & Participation

Class participation in lecture and utilize lab time with your TAs is strongly encouraged. Lecture will be used to provide skill overview and live demonstrations, many of while will kickstart your efforts for your project. Lab time will provide you access to equipment and the TAs for assistance.

Students are responsible for obtaining missed lecture content from other students in the class. All lecture slides/presented content will be made availabe online (Canvas/Gitlab), and lectures will be recorded via Panopto and posted to Canvas.

Participation on Ed Discussion is also encouraged, in the form of:

- Asking questions about the course material (ideally, publicly, so that others can benefit (Anonymous okay))
- Answering questions from other students
- Sharing interesting articles or resources related to the course material

Assignments & Grading

Grading

Fundamental knowledge will be assessed with online quizzes and the Nordic DevAcademy Fundamentals lessons.

Lab exercises will focus on implementing key functionality towards the final working device (a wireless ECG and temperature sensor). Each lab exercise will be evaluated with code reviews and technical reports that assess functionality.

Testing of the final working device will be presented in a final technical report.

All assignment grades will be posted to Gradescope/Canvas throughout the semester to track your performance.

Table 1: Grade Distribution

Grade Category	Relative Percentage
Quizzes	20%
Nordic DevAcademy Fundamentals Certificate	10%
Code Reviews	15%
Technical Reports	35%
Final Device & Report	20%

Extra Credit

Extra credit may be earned by completing:

- nRF Connect SDK Intermediate
- Bluetooth Low Energy Fundamentals

This can be used to offset late assignments, missed quizzes, etc.

Course Grade

This course is not "curved" (i.e., a distribution of grades will not be enforced), and a traditional grading scheme will be used (e.g., 90-93 = A-, 94-97 = A, 97-100 = A+).

Failing the course can happen with a cummulative score < 70 (C-).

> **Tip**
>
> Participation throughout the semester will influence rounding up/down for fractional grades.

Regrades

Any regrading requests need to be made **within one week of grades for a given assignment being released**. You must make the request via Gradescope and provide a description of why you feel a regrade is appropriate. Requesting a regrade could lead to additional loss of credit when an assignment is re-evaluated.

Some assignments will have an opportunity to be resubmitted based on grading feedback at the discretion of Dr. Palmeri.

Late Policy / SDAO Accomodations

Late submission windows will be available for all assignments, minus the final project, and should be used to accomodate acute illness, travel, high workload from other classes and other unforeseen circumstances. This late submission window can be utilized without penalty and without prior approval.

Students with SDAO accomodations for extended time on assignments can use this extended late submission window for all assignments.

Any assignments submitted after the late submission window will only be accepted for partial credit at the discretion of Dr. Palmeri or if prior approval was sought **before the original due date**.

Duke Community Standard

All students are expected to adhere to all principles of the Duke Community Standard. Violations of the Duke Community Standard will be referred immediately to the Office of Student Conduct. Please do not hesitate to talk with Dr. Palmeri about any situations involving academic honor, especially if it is ambiguous what should be done.

FAQ

Can I collaborate with other students?

Engineering is inherently a collaborative field, and in this class, you are encouraged to work collaboratively on your projects. That being said, all of the work that you submit must be generated by you and reflect your understanding of the material.

> **!** Important
>
> All resources used in your projects that were developed by another person or company must be properly acknowledged using comments in your code and lab reports.

Can I use AI?

The use of artificial intelligence is a rapidly developing resource / tool in engineering. In software development, there are many levels of AI-assitance available. Such form of assistance include the IntelliCode tools and GitHub CoPilot (free to students through the GitHub Education program). These tools can be leveraged to help with syntax.

> Caution
>
> You are strongly cautioned to not rely on these tools for logical implementation.