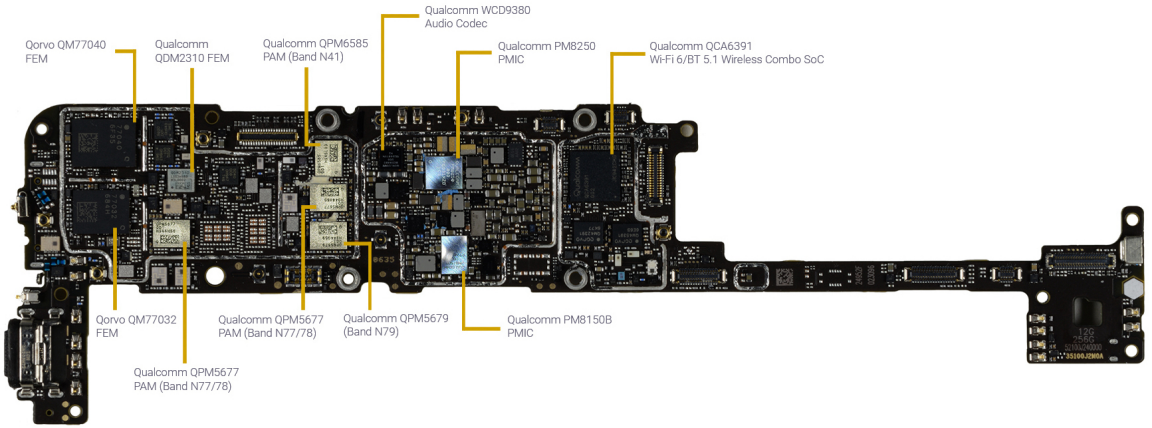


# What is this course?

Kizito NKURIKIYEYEZU, Ph.D.

## What is ETE 3261?

- A course on application of microcontrollers for embedded system design
- Focuses on hardware and software co-design
- Principles and practice of using Embedded RTOS
- Peripheral devices such as sensors and actuators to build a small embedded system.
- Designing of small real-world embedded system

## What is ETE 3261?

- A course on application of microcontrollers for embedded system design
- Focuses on hardware and software co-design
- Principles and practice of using Embedded RTOS
- Peripheral devices such as sensors and actuators to build a small embedded system.
- Designing of small real-world embedded system

## What you will learn?

I designed this course to teach you the theoretical and hands-on skills necessary to design real world embedded systems. By the end of this course, you will have developed your own embedded system from scratch, both hardware and software. Since we do not have a lab (yet!) we will unfortunately only use simulations

# What ETE 3261 is NOT?

- You are expected to have the necessary background. The important topic will be revised very quickly but it is your responsibility to catch up.

# What ETE 3261 is NOT?

- You are expected to have the necessary background. The important topic will be revised very quickly but it is your responsibility to catch up.
- Time commitment—Expect to spend more time self-learning

# What ETE 3261 is NOT?

- You are expected to have the necessary background. The important topic will be revised very quickly but it is your responsibility to catch up.
- Time commitment—Expect to spend more time self-learning
- Professionalism—You are adults and I will treat you as such

# What ETE 3261 is NOT?

- You are expected to have the necessary background. The important topic will be revised very quickly but it is your responsibility to catch up.
- Time commitment—Expect to spend more time self-learning
- Professionalism—You are adults and I will treat you as such
- Course content —The course involves lot of reading and self-studies

# PREREQUISITES

- I assume you understand the basic of digital circuitry, computer architecture and organization, ability to read and understand electronic schematics, competency in at least one high level programming language, hex and binary numbers, etc..

# PREREQUISITES

- I assume you understand the basic of digital circuitry, computer architecture and organization, ability to read and understand electronic schematics, competency in at least one high level programming language, hex and binary numbers, etc..
- You have the ability to read and understand unfamiliar topics

# PREREQUISITES

- I assume you understand the basic of digital circuitry, computer architecture and organization, ability to read and understand electronic schematics, competency in at least one high level programming language, hex and binary numbers, etc..
- You have the ability to read and understand unfamiliar topics
- You don't have to be an expert. But I don't expect you to ask me what's "flip-flop"? or what's "a bus"?

# Ideal expected knowledge

- **Computer architecture** —CPU Registers, memory addressing, data paths
- Electronics —Ohm laws, capacitor, diodes, resistors, transistors, op-amps, IC interfacing, analog filter design
- Good command of C (especially pointers in C)
- Fundamentals of data structures and algorithms

# EVALUATION STANDARDS

- QUIZ—There will be regular online quizzes over any material taught in the class to date. The quizzes should be easy if you learn the material as you go and to make sure you do not fall behind.
- EXAMS —UR's policy will be applied
- HOMEWORK
  - There will be several programming assignments
  - Each assignment expects design and debug some hardware simulation
  - ADVICE: Please try to work on this assignment early and ask questions if needed.

# EVALUATION STANDARDS

- QUIZ—There will be regular online quizzes over any material taught in the class to date. The quizzes should be easy if you learn the material as you go and to make sure you do not fall behind.
- EXAMS —UR's policy will be applied
- HOMEWORK
  - There will be several programming assignments
  - Each assignment expects design and debug some hardware simulation
  - ADVICE: Please try to work on this assignment early and ask questions if needed.

**NOTE** If you do not understand homework questions or if you do not complete your programming projects, odds are you will fail the exams and ultimately fail this class. Please make sure you understand what's on the homework and solve yourself the programming assignments. (and not memorizing them! It won't help.)

# ADVICE —Start working early

- Do the suggested exercises and make sure you write some code along the way. Review the course website and materials you received on the first day.

# ADVICE —Start working early

- Do the suggested exercises and make sure you write some code along the way. Review the course website and materials you received on the first day.
- Read the assigned materials for the first day ahead of time.

# ADVICE —Start working early

- Do the suggested exercises and make sure you write some code along the way. Review the course website and materials you received on the first day.
- Read the assigned materials for the first day ahead of time.
- We will be moving at a much fast pace. I will expect that you understand more deeply the concepts I teach in this course.

# What you will learn

- This course will make engineers out of you
- Team-based embedded-system design: design, document and implement embedded systems that solve a realistic biomedical system that include multiple realistic constraints.
- How to be an engineer:
  - Use your creativity creativity, resourcefulness and persistence to solve complex problem

# What you will learn

- This course will make engineers out of you
- Team-based embedded-system design: design, document and implement embedded systems that solve a realistic biomedical system that include multiple realistic constraints.
- How to be an engineer:
  - Use your creativity creativity, resourcefulness and persistence to solve complex problem
  - Extract information from obscure sources. You will read prodigiously. Your Labs will not have step-by-step instructions. This is typical for real design work.

# What you will learn

- This course will make engineers out of you
- Team-based embedded-system design: design, document and implement embedded systems that solve a realistic biomedical system that include multiple realistic constraints.
- How to be an engineer:
  - Use your creativity creativity, resourcefulness and persistence to solve complex problem
  - Extract information from obscure sources. You will read prodigiously. Your Labs will not have step-by-step instructions. This is typical for real design work.
  - This course will be hard. Some labs will take you more than 30 hours to complete. And you will, like any engineer, expected to produce high quality documentation (e.g., circuit design and specification circuit diagrams, PCB, documented C ode)

# What you will learn

- This course will make engineers out of you
- Team-based embedded-system design: design, document and implement embedded systems that solve a realistic biomedical system that include multiple realistic constraints.
- How to be an engineer:
  - Use your creativity creativity, resourcefulness and persistence to solve complex problem
  - Extract information from obscure sources. You will read prodigiously. Your Labs will not have step-by-step instructions. This is typical for real design work.
  - This course will be hard. Some labs will take you more than 30 hours to complete. And you will, like any engineer, expected to produce high quality documentation (e.g., circuit design and specification circuit diagrams, PCB, documented C ode)
  - Read and understand long datasheets

# What you will learn

- This course will make engineers out of you
- Team-based embedded-system design: design, document and implement embedded systems that solve a realistic biomedical system that include multiple realistic constraints.
- How to be an engineer:
  - Use your creativity creativity, resourcefulness and persistence to solve complex problem
  - Extract information from obscure sources. You will read prodigiously. Your Labs will not have step-by-step instructions. This is typical for real design work.
  - This course will be hard. Some labs will take you more than 30 hours to complete. And you will, like any engineer, expected to produce high quality documentation (e.g., circuit design and specification circuit diagrams, PCB, documented C ode)
  - Read and understand long datasheets
  - Apply material from courses you have already taken

# What you will learn

- This course will make engineers out of you
- Team-based embedded-system design: design, document and implement embedded systems that solve a realistic biomedical system that include multiple realistic constraints.
- How to be an engineer:
  - Use your creativity creativity, resourcefulness and persistence to solve complex problem
  - Extract information from obscure sources. You will read prodigiously. Your Labs will not have step-by-step instructions. This is typical for real design work.
  - This course will be hard. Some labs will take you more than 30 hours to complete. And you will, like any engineer, expected to produce high quality documentation (e.g., circuit design and specification circuit diagrams, PCB, documented C ode)
  - Read and understand long datasheets
  - Apply material from courses you have already taken
  - Ability to communicate your work succinctly and precisely

# What you will learn

- You will learn how to develop medium complex embedded system from scratch
- The objective is to replicated code that can be run on an Arduino[1] using ANSI C[2] on an ATmega328P microcontroller
- LAB—Not sure how this will go! UR has no lab equipment and no lab time has been allocated for this course
- It is hoped that, by the end of this course, you will know how make a embedded system product from scratch.

[1] https://en.wikipedia.org/wiki/Arduino

[2] https://balau82.wordpress.com/2011/03/29/programming-arduino-uno-in-pure-c/

# The end