Embedded Systems (LS00DS51-3002) Exercises Spring 2026

TA: Alexander Beattie (alexander.beattie@uef.fi)

February 4, 2026

Repository: `https://github.com/gateway240/embedded-system-uef`

1 Timetable

#	Date	Day	Time	Room
1	13.02.2026	Fri	14:00–16:00	SN204
2	20.02.2026	Fri	14:00–16:00	SN204
3	27.02.2026	Fri	14:00–16:00	SN204
4	06.03.2026	Fri	14:00–16:00	SN204
5	13.03.2026	Fri	14:00–16:00	SN204
6	20.03.2026	Fri	14:00–16:00	SN204
7	27.03.2026	Fri	14:00–16:00	SN204
8	10.04.2026	Fri	14:00–16:00	SN204
9	17.04.2026	Fri	14:00–16:00	SN204
10	24.04.2026	Fri	14:00–16:00	SN204
11	29.04.2026	Wed	14:00–16:00	SN204
12	08.05.2026	Fri	14:00–16:00	SN204
13	13.05.2026	Wed	14:00–16:00	SN204
14	22.05.2026	Fri	14:00–16:00	SN204

2 Course Goals

After completing the course, the student:

- Understands the architectures of common embedded systems
- Capability to choose the appropriate modules and components for a given problem
- The skills to build, test and verify embedded systems

The course develops the following generic skills:

- digitalization
- leadership and development
- sustainability and responsibility
- critical thinking
- identification and development of expertise
- interaction and communication

2.1 Content

1. Introduction to Embedded AI System architectures
2. Electronics and logic gates
3. Embedded input, output and timers
4. GPIO Protocols, I2C
5. SPI, UART Interrupts Control Interaction
6. Peripherals Testing and verification
7. IOT, Internet protocols
8. Embedded system optimization - Quantization
9. Future trends

3 Exercises

You can work by yourself or in a pair. Being physically present at the exercise counts as 1 point. The remaining 6 points are earned as described below. It is possible to earn 7 points per week (total 98 points). 50 points are required to pass the exercises. In the exercises, you will create your own sensor—an inertial measurement unit (IMU)—from scratch!

The exercises feature a progression of microcontrollers. At the beginning the Arduino Nano Every contains an ATMega4809 chip which you should be familiar with from the Arduino UNO. Next, the Arduino Nicla Sense ME features a more advanced nRF52832 32-bit Arm Cortex-M4 processor, and finally, the BeagleBoardY-AI microcontroller features an AM67A 64-bit Arm Cortex-A53 CPU and an Arm Cortex-R5 subsystem.

3.1 Exercise 1 - Introduction

Review of concepts from embedded systems. Start with an Arduino Nano Every. Introduce final minimum viable product (MVP) project and course goals. Solder headers and components if necessary.

Tasks:

1. (1 point) Connect a serial monitor, create an echo hello world program
2. (2 points) Create a variable speed DC fan using a transistor, potentiometer, 9V battery, and a PWM signal from the arduino. Accept serial input values to control the fan speed. Reference: `https://www.hackster.io/onedeadmatch/control-dc-motor-with-npn-transistor-arduino-pwm-cdaf2e`
3. (3 points) Program your own motor control profile with at least 5 distinct speeds.

3.2 Exercise 2 - Accelerometer

Build an IMU with only an accelerometer

Tasks:

1. (1 point) Connect the accelerometer and display the readings over serial output
2. (2 points) Connect the OLED display and display the raw accelerometer readings on the OLED display
3. (3 points) Compute orientations from the built-in library and display them on-screen

3.3 Exercise 3 - Gyroscope

Add a gyroscope! Design and fabricate a housing for both sensor which will fit all 3 sensors and the arduino (e.g. 3D printed or hand made) so the sensors move as one unit

Tasks:

1. (1 point) Devise a mathematical solution for combining the accelerometer and gyroscope readings to estimate orientation
2. (2 points) Create the sensor housing and add the accelerometers and gyroscopes
3. (3 points) Connect the gyroscope to the arduino and display the readings over serial output

3.4 Exercise 4 - Magnetometer

Add a magnetometer! Provide FPS benchmarks as well as orientation estimates.

Tasks:

1. (1 point) Update the mathematical solution to include the accelerometer, gyroscope, and magnetometer readings to estimate orientation
2. (2 points) Add the magnetometer and display readings from all 3 sensors
3. (3 points) Add an FPS calculation for the sensors using a rolling average (to prevent jitter) and display the value on-screen

3.5 Exercise 5 - IMU Math Applied

Make your own formulation for the IMU (Kalman filter, least squares, etc.) and implement it on the device. IMU Math and Software implementation Derive equations for IMU (magnetometer, gyroscope, accelerometer) sensor fusion. Implement the algorithm in real time on the microcontroller.

Tasks:

1. (1 point) Implement your derived mathematical formula in code that runs on the Arduino Nano Every
2. (2 points) Demonstrate your solution works by outputting quaternions on the screen
3. (3 points) Create a data logger which records the time series data to your computer and visualize the output

3.6 Exercise 6 - Arduino Nicla Sense ME

Introduce the Arduino Nicla Sense ME and develop firmware!

Tasks

1. (1 point) Demonstrate working Zephyr firmware cross-compiled from your own machine with the echo hello world task
2. (2 points) Display quaternion values from the on-board sensors on-screen with the new sensor
3. (3 points) Design a test protocol, log values from your homemade sensor and this new sensor and compare the results (graphically)

3.7 Exercise 7 - MVP Start

Minimum viable prototype (MVP) work begins! Select a problem that challenges you and is interesting to you that can be solved with embedded systems. Design a minimum viable product (MVP) and present a development plan. This needs to include empirical, quantitative results and benchmark steps. Create tasks for the individual project work weeks.

Tasks:

1. (1 point) Build upon the existing sensor by selecting a problem that challenges you and is interesting to you that can be solved with embedded systems
2. (2 points) Design a minimum viable product (MVP) and present a development plan
3. (3 points) Demonstrate control results against the current sensor that you will use to benchmark your design against (in a graph)

3.8 Exercise 8 - Battery, Wireless, and IO

Calculate battery capacity for wireless applications. Setup a wireless communication link (Bluetooth, etc.) between the Arduino and a collection node.

Tasks:

1. (1 point) Calculate an estimate for the necessary size of your battery, given desired runtime parameters (e.g. it can run for 8 hours). Connect the provided battery to the Arduino Nicla Sense ME and demonstrate sensor operation wirelessly to your computer.
2. (2 points) Add a button which can be used to toggle settings in the sensor. Create code that will print to serial out when the button is pushed.
3. (3 points) Benchmark the battery life of your embedded solution with figures

3.9 Exercise 9 - Embedded AI with BeagleBoardY-AI

AI topics! Implement a collector node on the BeagleBoardY-AI. Create and train a simple neural network for a task related to the sensor data (e.g. sensor fusion, magnetometer-free, etc.)

Tasks:

1. (1 point) Setup your BeagleBoardY-AI and demonstrate that you can ssh in and execute a hello world program
2. (2 points) Setup Bluetooth on your BeagleBoardY-AI and test that a sensor can connect
3. (3 points) Implement a data collection node for your developed sensor and show that data is properly received and logged to a file

3.10 Exercise 10

Solution analysis and presentation. Demonstrate your solution. What went wrong? What went right? How would you improve the solution the next time around?

Tasks:

1. (1 point) Present something, regardless of it it works
2. (2 points) Show results from a neural network running on the BeagleBoardY-AI and compare them (in a graph) to actual math and the on-board sensor fusion. Present a fully working solution with detailed graphs and comparisons
3. (3 points) Design a protocol to handle the simultaneous Bluetooth connection problem

3.11 Exercise 11

MVP fabrication phase 1. Begin working on the MVP and present your progress. Remember not to spend too much time getting stuck on one feature. Maybe you can skip it all together?

Tasks: Specified in individual project plans

3.12 Exercise 12

MVP fabrication phase 2. The bulk of the work for the MVP happens here. You will be surprised by the road blocks you encounter here. Don't be afraid to ask for help from me or your peers. Push through the adversity and you'll be happy with the result.

Tasks: Specified in individual project plans

3.13 Exercise 13

MVP fabrication phase 3. Testing, Quality Assurance, and Validation Test your prototype with edge cases and unexpected events. Benchmark and demonstrate success.

Tasks: Specified in individual project plans

3.14 Exercise 14

MVP pitching competition! Present your MVP. Demonstrate benchmarks to why and how it works? What makes it valuable and why should we invest in it?

Tasks:

1. (6 points) Pitch your idea successfully