EEE3096S: Embedded Systems II

LECTURE 16:

GA TEST PREPARATION

Presented by:

Dr Yaaseen Martin

Electrical Engineering
University of Cape Town

GA5 TESTING

- The main GAs in this course will be assessed through venue-based test writing.
- The test questions comprise a combination of text, sketches, and possible calculations to assist in explanations.
- You will need to be familiar with some of the course concepts, such as SPI/I2C, cross-compilers, etc — but the test is primarily **design focused**.

GA5 TESTING

This GA assignment assesses your competence in understanding and how you would apply a selection of engineering skills relevant to engineering work that may be carried out by graduates of either the Mechatronics programme or the Electrical and Computer Engineering programme. The specific categories, as used in the ECSA documentation, are as follows:

B. Skills (Application of Knowledge)

B.1 Embedded System design as a complex process.

B.2 Use of design diagrams including (for relevant to the development and use of embedded systems) Finite State Machine (FSM) & Algorithmic State Machine (ASM)

B.5 Use (cross-)compiler for developing ES software

B.6 Use of standard embedded systems communication protocols

GA5 ASSIGNMENT TEST PREPARATION

- The worksheet 'EEE3096S 2023 GA Assignment.pdf' involves problem-solving tasks to help prepare you for the topic and types of questions that may be asked in the GA test.
- Understanding the assignment design concept will much facilitate your understanding of the broader design problem that the test questions will be asking further, more specific, questions about.
- It is planned that these activities will also connects towards the mini-project.
- Assignment PDF is available on the Amathuba site:

Content/Class Tests/2023/EEE3096S 2023 GA Assignment.pdf

GA5 ASSIGNMENT TEST PREPARATION

- Your preparation of working through the tasks can be done in teams/groups, where you and a few classmates can try to tackle the problems and exchange strategies or ideas.
- But **note:** the test-writing is **individual** — not teamwork!
- GA tests will be **closed-book**.

DO YOU NEED TO PASS THE GA TEST QUESTIONS?

- The GA test is a requirement for **all** students registered for the course and it **will** count towards your final mark.
- You do **not** need to pass the GA test if:
  - You are not in an engineering programme, or
  - You have passed GA5 in another course; in this scenario you do not need to pass the GAs again, but you **still** need to attend and write the test for DP.
- Passing the GAs
  - EEE students must pass all the GA questions (for ECSA purposes) in the written test; Science students are allowed to fail 🤷‍♂️
  - A second-attempt GA test will be provided for students who fail to meet any of the GAs; only the failed GA(s) will be reassessed and your original mark will not be changed regardless

DO YOU NEED TO PASS THE GA TEST QUESTIONS?

- Second attempt test:
  - A second attempt for all the GA test questions will be provided later in Term 4
  - If you are absent for the first test, you **must** write the second test
- Science students who fail any GA question will **not** write the second-attempt test as they do not need the GAs for DP

GA TEST TOPIC: LIGHT-OF-THINGS (LOT) SENSOR

The GA assignment relates to approaches and plans about design, tools to use, development approaches, and performance metrics related to Light-of-Things receiver and transmitter nodes that you are expected to be able to construct.

Note: You will **not** be physically constructing anything as part of this work; just thinking about (and planning) the design of the system!

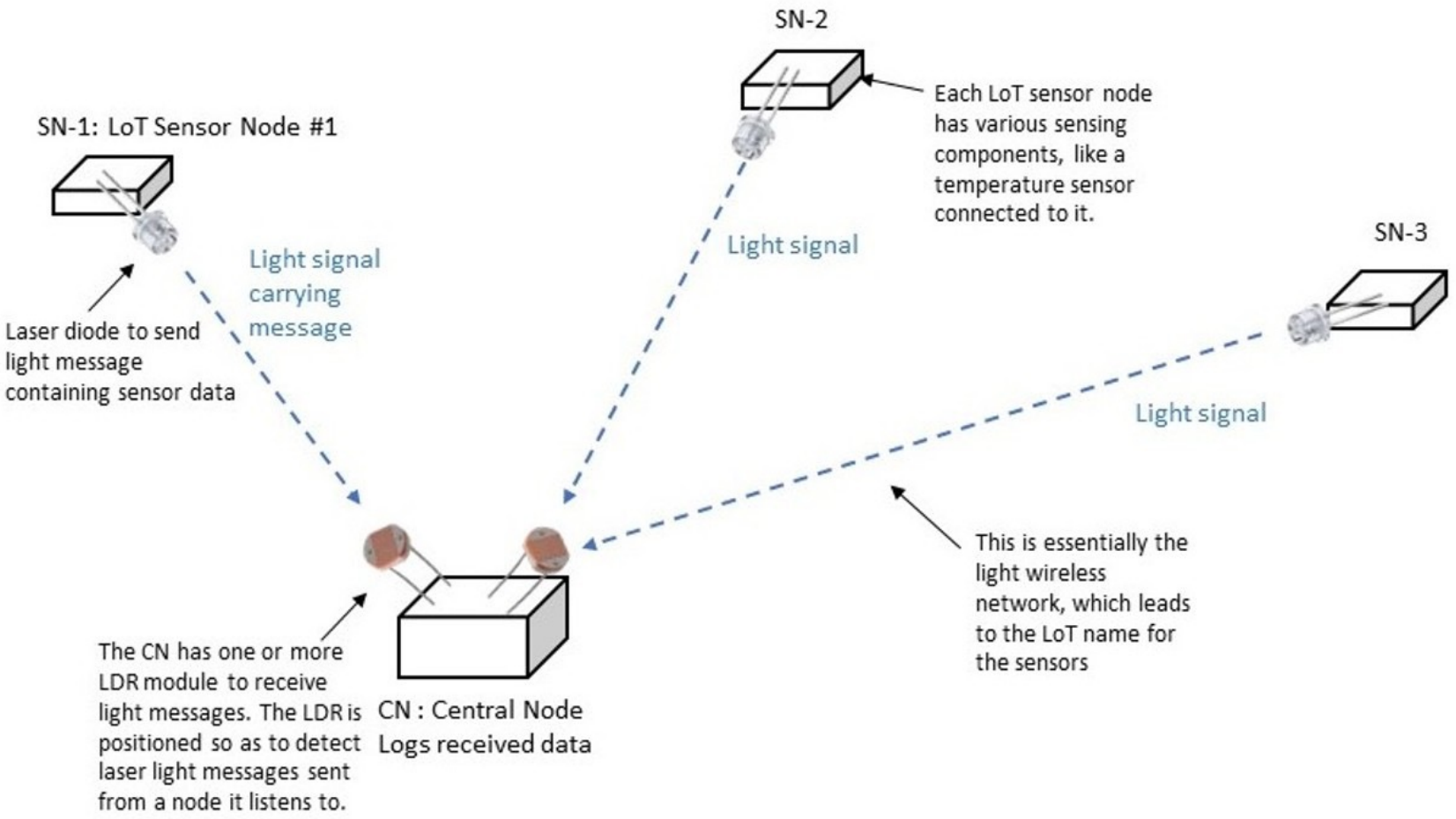
THE APPROACH TO FOLLOW

The approach to handling this assignment is as follows:

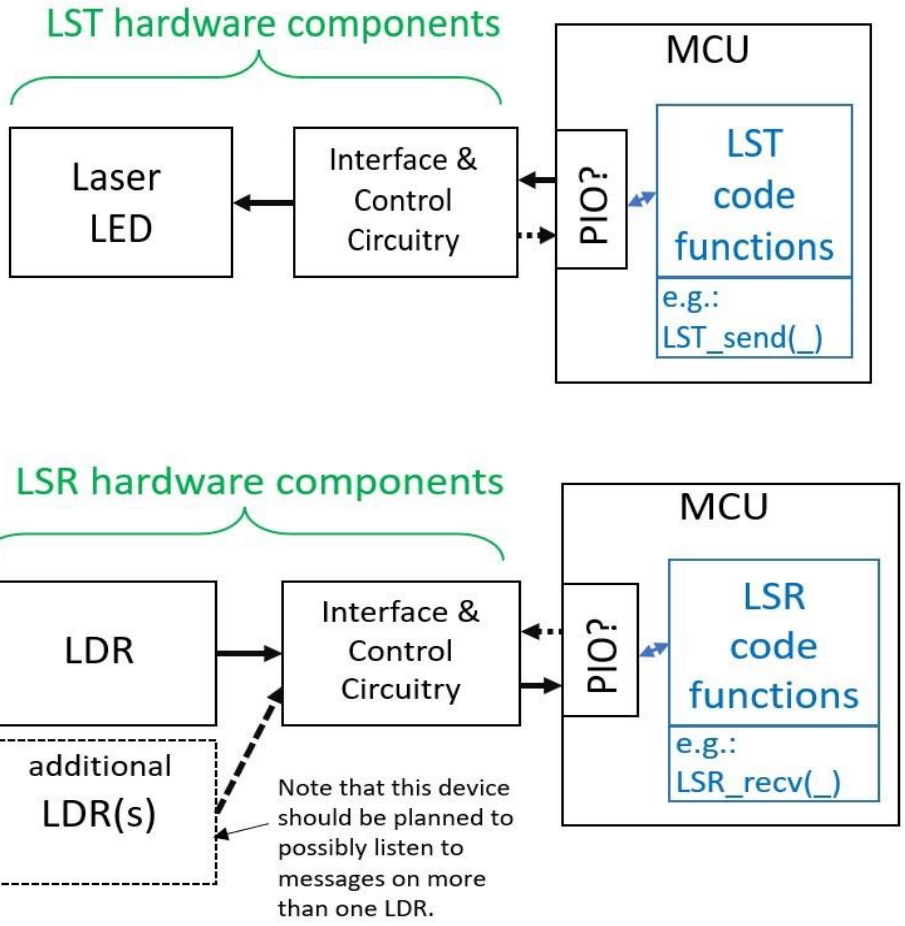
1. Read over EEE3096S 2023 GA Assignment.pdf as it describes the LoT transmitter and receiver system requirements in detail.
2. Review the requirements for the Embedded Software API needed for the LoT modules.
3. Do the readings provided in the table at the end of the assignment document.

This assignment does not require physical carrying out of the development tasks; it instead focuses on a theory and understanding what development tasks are to be done and explaining how these tasks could be carried out.

LIGHT-OF-THINGS (LOT) SENSOR DATA FEEDS



LIGHT-OF-THINGS (LOT) SENSORS



API AND PROTOCOLS

READ

- **Definition:** Application Programming Interface (API):

"…an API lists a bunch of operations that developers can use, along with a description of what they do. The developer … just need to know [functions] available for use in their app."

- Part of the GA assignment asks you to think about what the API for the sensor node (for sending data) might look like, and what the API for the central node (for receiving data) might look like. You might decide to have just one API and the compiler decides which functions to compile object code for depending on which side it is used for.

- Think about communication protocols as well as the concept of parity bits, or Manchester encoding, for validation checking

EXAMPLE – DECIDING A METRIC FOR LOT

READ

Identifying the constraining factors in your design…

There are various standard performance metrics.

For this assignment, you can motivate for using one or more standard metrics, such as the Baud rate of communication on a perfect channel.

Or, and possibly in addition, you could develop and motivate for your own metrics that would be useful in concisely expressing characteristics of your system.

Let's go though an example of one such case…

EXAMPLE – DECIDING A METRIC FOR LOT

READ

One such factor is **how far communications data can travel** via the light signals. This is impacted by both the laser LED used, such as it's focus and the power of the beam available. It is also impacted by the environment.

Accordingly, it can be useful in expressing the range that the system can provide. This range could be considered straight line-of-slight, the beam of light traveling from the LED. But it is not just how far the light beam will travel! The signals need to be received and demodulated in some way, by the receiving node. The efficiency and accuracy of this receiver is impacted in various ways and dependent on the hardware utilized. This, for this system the notion of range depends on, at least, both the transmitter and receiver design. From some conceptual exploration, we can introduce a metric of 'maximum workable detection limit' for the system, in which data can reliably be transmitted and received for the system.

EXAMPLE – DECIDING A METRIC FOR LOT

READ

Definition of 'maximum workable detection limit' (MWDL):
MWDL is expressed in the unit of meters. It indicates the maximum distance between the transmitter and receiver for communication to operate at least 80% reliably or a BER* of 0.2. This range is restricted to line-of-sight communication between the LoT transmitter and LoT receiver.

Additional limiting factors can be identified, e.g.
MWDL is dependent on the environment in which the communications is done.

* BER = Bit Error Rate

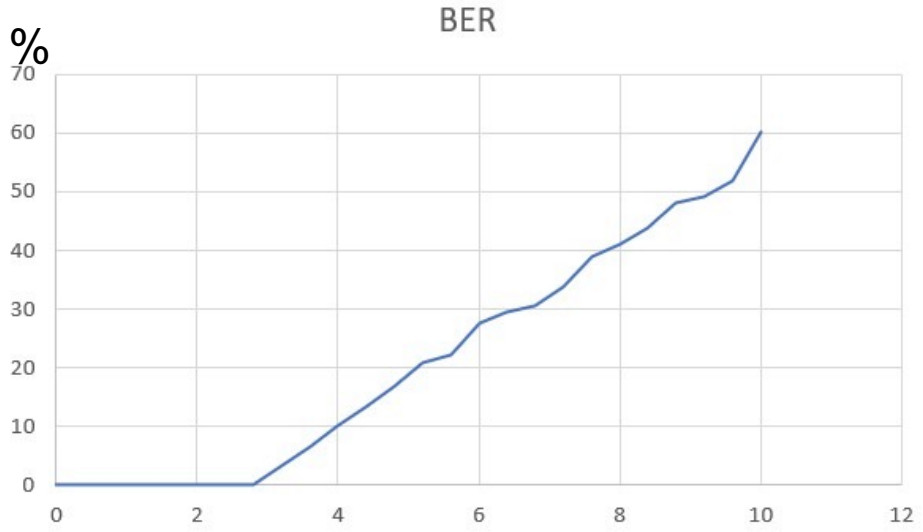
EXAMPLE – VISUALISING A METRIC FOR LOT

READ

A useful means to visualize the MWDL can simply be a measure for anticipated accuracy drop-off beyond the range at which the communications starts becoming less reliable.

This probably not going to be an exponential drop-off, unlike un-directed radio, as it is a focused directional beam that we are dealing with. This it will probably be a linear drop-off, but the gradient of that drop-off may be of particular use to know how far, beyond the MWDL, some of the data that is sent can still be recovered.

Concept of visualisation for comms degradation beyond MWDL



HINTS

- Be able to draw design diagrams such as flowcharts, block diagrams, FSM, etc.
- Brush up on your embedded comms (I2C, SPI, RS-232, etc) and know the differences between them and how they are used
- Read up on parity bits, Manchester encoding, and potentially other means of doing error checking on data packets
- Think about the functionality your API should have, and what its purpose would be in the system design
- Understand the purpose and definitions of cross-compilers, toolchains, debuggers, development/execution/runtime environments, Big/Little Endian, etc.
- Know how an ADC works (to be formally covered early next term) and the role it plays in this system design
- Use tomorrow's quiz as a means of gauging your understanding of the course material

QUIZ TOMORROW (29 AUGUST)

- Not for marks; not compulsory
- 45 minutes long
- Format:
  - MCQ
  - True/False
  - Short + long answers (written)
  - Diagrams
- Scope:
  - Everything covered in Term 3
  - Focus on Embedded Comms + ARM processing
- Quiz will be uploaded to Amathuba after the Wednesday lectures
- Solutions will also be worked through during the second period
- Final Q&A to wrap up my set of lectures

CONFERENCE OPPORTUNITY

- Prof Winberg is looking for around **two** EEE3095/6S students to volunteer to assist with a one-day conference, ICECET 2023: http://www.icecet.com/committees
- The conference is in **Cape Town** at the Captonian Hotel this year, taking place on **16 November**
- Free meals, refreshments, etc. will be provided during the conference
- Your responsibilities would include: assisting at the registration desk, checking venues, helping delegates set up slides, attending some of the sessions (if interested), etc.
- Good networking opportunity and learning experience if you are interested in research; and could add it to your CV
- Email Prof Winberg if interested: simon.winberg@uct.ac.za


Fin.