EEE3096S GA5 Assignment 2023

Graduate Attribute (GA) 5 Part B:

Understanding the Application of Engineering Knowledge

Synopsis

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

1 Introduction

This assignment aims to assess your understanding of engineering approaches and the skills needed to carry out development tasks; this assessment is done through a process of posing a 'representative design problem' (albeit a small one), for which you are tasked with understanding the problem and to propose effective methods and solution strategies for accomplishing the development needs.

This year, the 'representative design problem' relates to the **Light-of-Things (LoT) sensor** project that is worked on in Term 4 (**only for CS students**). This assignment relates to transmitter and receiver modules to be considered for the LoT messaging, whereby light signals are sent to receivers from a transmitter or set of transmitters.

The approach to handling this assignment is as follows:

1. Read over the description of the LoT transmitter and receiver modules, and associated LoT network, as described below.
2. Review the requirements for the embedded software Application Programming Interface (API) needed for LoT modules.
3. Read over the following, links given in the table at the end of this document:
    a. Reading concerning aspects of the laser diodes/LEDs and light sensors, and mirrors
    b. Manchester Encoding and the use of parity checking
    c. Read up on what an API is

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

Assessment of these tasks will be done using written (paper-based) independent testing in a test venue. The test questions will pose a design task related to the problem discussed in this document and the knowledge required to carry out such a task. **Two** attempts at the test questions will be allowed, and you will need to achieve an adequate pass mark for each test question to be provided the GA for this course; otherwise, your **DP will be withheld**. The main test, **Class Test 1**, will constitute your first attempt of the questions for the GA assessment; there will also be a second attempt for the test for students who do not pass the first time.

Teamwork options: It is recommended that you work in **teams of three or four** (where possible) on this assignment. The principle is really for you to help each other understand, think about, and consider solutions for the development tasks discussed. **However**, you will each be **individually assessed** in closed-booked testing of your understanding and responses to test questions related to this assignment.

2 Design Concept: The LoT Data Transfer Network

This assignment concerns development considerations for a 'Light of Things' — abbreviated as LoT in this document — sensor data transfer network. The baseline version, and the version that you are mainly going to be considering, utilises **simplex** communication in which a transmitter node sends data to a receiver node and there is no mechanism provided by which the receiver can get data or acknowledge back from the transmitter using the light communication links.

The Figure 1 below provides an example scenario of how such a network might be set up. In this scenario, there is a receiving central node (labelled CN) to which logged data messages are sent via light beams. The transmitting sensor nodes (labelled SN-1 to SN-3), sample one or more physically attached sensors (in this case a microphone and light reader) and transmit this sensor data by light signals back to the CN.

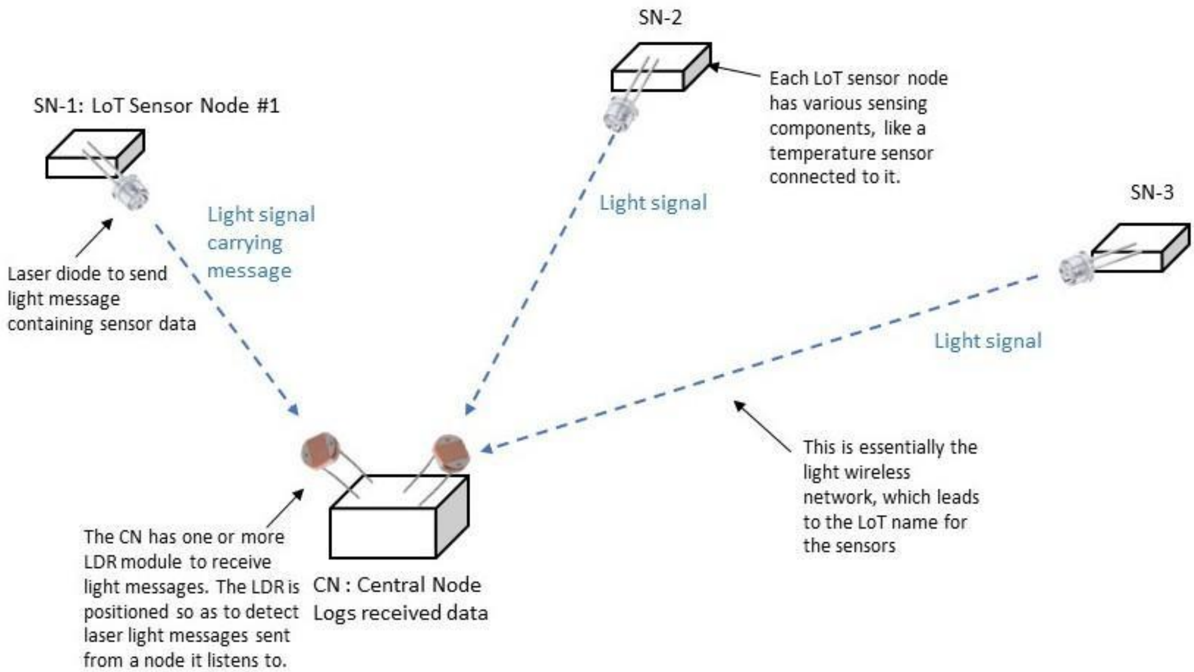

Figure 1: IoT laser light data transfer network

The sensor data to send by a node is first packaged into a message package, and this is appropriately encoded and transmitted by the sensor node's laser diode/LED transmitter. In the standard configuration, each sensor node needs its 'light signal transmitter' (LST) — mainly composed of a laser diode/LED — to point directly at a Light Dependent Resistor (LDR) of the CN's 'light signal receiver' (LSR). LSR can have one or more LDR from which it can receive messages. Part of the problem is deciding an effective way that the LSR can be implemented; should it just listen to the LDR that is giving the highest voltage (from its signal conditioning circuit), or should it have some means to keep listening to just that LDR from which a valid incoming signal was first detected?

Probably an immediate first concern you may have on this proposed problem — and if you correctly assumed the signals are just being sent through the air — is what lighting levels could this system work at, and how far apart could the transmitter and receiver be? These are things that you will need to consider in the upcoming development thinking tasks, and you may be expected to respond to this in test questions. Other likely questions are: how fast is the data transfer, and what if two transmitters (pointing to the same LDR on the CN) are trying to send at the same time?

3 Design Tasks

You need to go about designing the aspects as described below. Note that some of these aspects are hardware related (components, circuitry needs, hardware interfacing), and other aspects are more software related. Most of these tasks are more thinking and planning aspects, considering how you would go about instrumenting solutions; you are not actually required to physically implement the solutions. ***Note:*** Later in the course, as part of the project for CS students, you may need to implement some of these aspects.

3.1 Design Task 1: Design of the light signal transmitter (LST) device

The LST is not merely a laser diode. More parts, both in terms of hardware and software, are needed to implement the device. You need to think about this and consider some ideas for what such a device would incorporate. A high-level view is shown in Figure 2, indicating that (besides the LED) some control circuitry would be needed to connect the microprocessor to run LST code functions (note that the labelled MCU refers to the microcontroller, i.e., the STM that you might use).

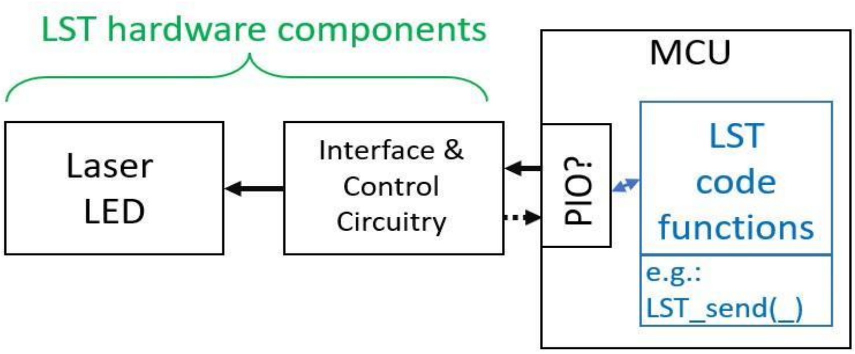

Figure 2: Light Signal Transmitter (LST) hardware and code module

For this task, you need to respond to the points listed below.

To do:

- Provide a sketch for the circuit you would use to connect your LED to the MCU.
- Discuss how you would design the software interface that will be used to control the LST hardware from the MCU.
- The LST will use an ADC to sample sensors (which may be multiplexed) on the node, and this data may be packaged into a data package (see later) and sent by light signals to the CN for

logging. Make sure to know how you would go about connecting to an ADC on your MCU and how a single ADC might be multiplexed between (potentially multiple) physical sensors.

- Propose a brief API for the LST, indicating the commands and any data types provided by the API.

3.2 Design Task 2: Design of the light signal receiver (LSR) device

The LSR device is very similar in a high-level view to the LST, as illustrated in Figure 3.

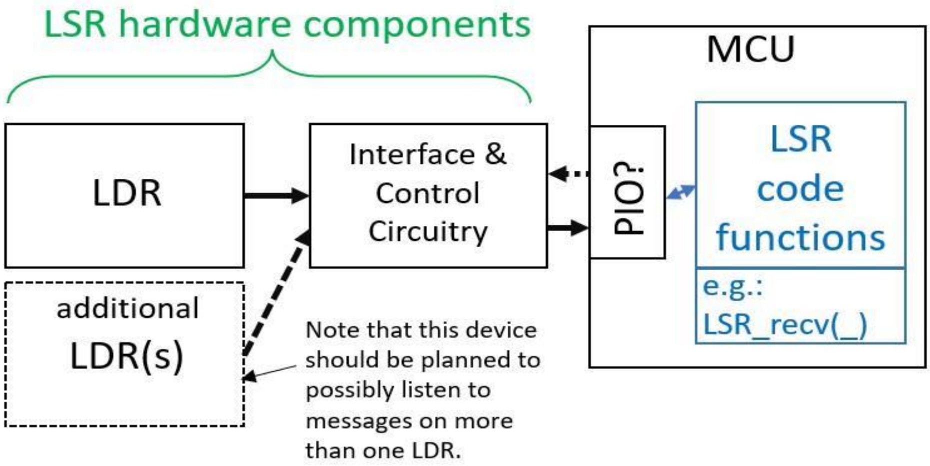

Figure 3: Light Signal Receiver (LSR) hardware and code module

To do:

- Provide a sketch for the circuit you would use to connect your LDR to the MCU.
- Discuss how you would design the software interface for the MCU to control and receive data from the LSR hardware.
- Develop a finite state machine (FSM) to express visually how the LSR will work.
- Propose a brief Application Programming Interface (API) for the LST, indicating the commands and any data types that are provided by the API.

3.3 Design Task 3: Communication Protocols and API functions

The baseline system for this LoT network allows only simplex communication, where a sensor node will only be able to send out light messages and will not be able to receive any acknowledgements or data back via the light channel. If some sort of acknowledgement is needed for a system implemented using this network, that acknowledgement would need to be provided in some other way, e.g., via an audio signal for which the sensor would need to have components to listen for such signals.

For now, you can assume that the user will be responsible for positioning and adjusting the laser LEDs on the LST devices, as well as adjusting the LDRs of the LSR device of the central node.

To do:

- Assume the LST will send out data as a stream of light pulses. The intensity of these light pulses is fixed according to the circuit you will use (i.e., you cannot change the light intensity of a pulse sent out to send more data in a single pulse).
- Propose a simple packet structure and communication protocol for how you would package and transmit a collection of data bytes. Assume that in some cases you may want to send just short messages, like “sensor active”, and in some cases longer messages, such as the time of a sensor reading and the sampled data that was obtained from the sensor.

- Considering that the sensor node will be obtaining data by reading an ADC, consider if this ADC can be multiplexed to more than one physical sensor.
- Provide a Sensor Node Application Programming Interface (SN-API) that is in addition to the API for the LST, providing commands for reading sensor data that then puts sampled data into a package that will be sent to the central node. Propose API commands and methods by which you can provide commands to send the constructed package over to the central command, and logging (on the sensor node) that the package was sent, and when. Each SN has a sensor number used by the central node to keep track of where sensor readings originated.
- Provide a Central Node Application Programming Interface (CN-API) that is in addition to the API for the LSR, that provides commands for listening for and receiving packages sent from sensor nodes — and adds that package to a record of sensor data packages received from that sensor node. Propose API commands and methods by which you can provide commands to recover packages sent from a sensor node to the central node, and commands for accessing the recorded data packages and the data stored in them. Note that the packages should contain time stamps as well that can be accessed by a relevant CN-API command.

3.4 Protocol considerations and assigned reading

When considering the protocol for packaging and sending data from the LST module, know that you may be asked to incorporate parity bits, Manchester encoding, or an equivalent means for improving data integrity and its ability to 'self-synchronise' — which eliminates the need for a shared clock or additional clocking mechanisms to know what length communication pulses are expected to be.

Consider how, in your API design for the SN and the CN that you may want to have an option (e.g. a runtime flag) that will allow the node to be switched to using Manchester Encoding when sending out, or listening to, data pulses. Be sure to tead up on the concept of Manchester encoding as this could be potentially useful for your design.

Use of a parity bit should also be considered in designing the way for checking the validity of bytes that are sent in the message packages that are transmitted by the LST. Build into your protocol the possibility of supporting the following four types of parity or byte validation checking; you can refer to these options as the Message Type option for the way that parity checking is done in data packages sent by your LST, the options are:

Message Type (MT) options:

0 = even parity (not of 1's + parity bit must be even)

1 = odd parity (not of 1's + parity bit must be odd)

2 = space (parity bit is always 0)

3 = mark (parity bit is always 1)

4 = no parity bit included (the next 8 data bits follows immediately)

You should be able to discuss the pros and cons of these different parity options, why you might want to include them, if you think it would be recommendable for all message types or only some. You are likely to be asked to reflect on these points in the GA test. You may want to consider proposing an alternate error checking approach — which you and your teammates can either make up on your own — and think about possible pseudocode for how such checking might be done.

4 Preparing for the GA test

Clearly, the tasks mentioned in this document show how the development of these aspects (that constitute parts of embedded systems) can be a complex process. In reading over this assignment, make some notes on which aspects may be difficult and which design options could have a significant effect on the effectiveness, speed, and power usage of implementing the proposed LoT network; note that this last point connects with item B.1. of GA 5.

5 Reading List

Be sure to brush up on your coursework — particularly the sections on Embedded Communications, parity bits, cross-compilers and toolchains, and Finite State Machines and ADCs (both covered at the start of Term 4). In addition, the table below indicates the readings to complete and topics that you should know about to be well prepared for the GA test:

Topics	**Description**	**Link**
Concave Mirrors and Convex Mirrors	This discusses issues of mirrors that can be used to reflect light beams and conversion or diversion of the beam	https://byjus.com/physics/concave-convex-mirrors/
About reflecting light	Explanation of how light can be reflected with lenses and mirrors	How mirrors, lenses, and prisms shape light systems | Laser Focus World
Manchester Encoding in Computer Network	Thorough but easy to follow explanation of Manchester encoding and uses in computing and comms	Manchester Encoding in Computer Network - GeeksforGeeks
Laser LED datasheet	An LED may also work if a laser diode is unavailable	Laser LED datasheet
LDR datasheet	Standard	LDR datasheet
APIs	Brief explanation of APIs and why they are used.	What is an API - GeeksforGeeks