**CG1112 Engineering Principles and Practices II**
**Tutorial 5**

## Section 1: Protocol Design

1. When Alex's Raspberry Pi communicates with his Arduino, we populate a variable of type TPacket, then serialize the TPacket data structure simply by copying over the entire structure into an array of char (i.e. an array of bytes):

```
TPacket packet;

… Code to populate packet's fields …

char buffer[MAX_PACKET_LEN];
memcpy(buffer, &packet, sizeof(TPacket));
```

However when Alex communicates with another host over the Internet through alex-server.cpp and alex-client.cpp, he populates an array instead of a structure, using a format similar to the following, where each element is of type int32_t.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 3 | Command | Param 0 (4 bytes) | | | | Param 1 (4 bytes) | | | |

a. Why do you think this was done in place of serializing a data structure? What are the relative advantages and disadvantages of each approach?

b. Given your answer above, suggest how we could have implemented data structure serialization on alex-client.cpp and alex-server.cpp.

## Section 2: Secure Networking

1. In Week 8 Studio 2 it was mentioned that in order for your certificates to be accepted on web browsers, they must be signed by a recognized Certificate Authority, and not by yourself. Explain how web browsers enforce this requirement.

2. In early March 2018 certificates issued by Trustico on behalf of Digicert for 23,000 websites were revoked when the CEO of Trustico emailed the private keys matching these certificates to Digicert. Why was such action taken to revoke the certificates?

## Section 3. SLAM

1. [Is that a wall?] It is quite cool to see Hector SLAM in action. Instead of the multiple LiDAR points we see in W8S1, solid lines representing real world obstacle are drawn instead. Let's take a look at how such "line segment" can be deduced from data points computationally.

   First, take a look at the basic idea of RANSAC (RANdom SAmple Consesus) at wikipedia https://en.wikipedia.org/wiki/Random_sample_consensus . The overview section is pretty high level and sufficient for our discussion.

   The Wikipedia page focus on getting a single "best fit line" through the points. However, in a typical LiDAR scan, there are multiple lines, each representing some parts of a landmark. Let us adapt the RANSAC algorithm as follows:

While

1. There are still unassociated LiDAR readings,
2. **and** the number of readings is larger than the consensus,
3. **and** we have done less than **N** trials.

Do:

- Select a random laser data reading.
- Randomly sample **S** data readings within **D** degrees of this LiDar data reading (for example, choose 5 sample readings that lie within 10 degrees of the randomly selected laser data reading).
- Using these **S** samples and the original reading calculate a least squares best fit line.
- Determine how many laser data readings lie within **X** centimeters of this best fit line.
- If the number of laser data readings on the line is above some consensus **C** do the following:
  - Calculate new least squares best fit line based on all the laser readings determined to lie on the old best fit line.
  - Add this best fit line to the lines we have extracted.
  - Remove the number of readings lying on the line from the total set of unassociated readings.

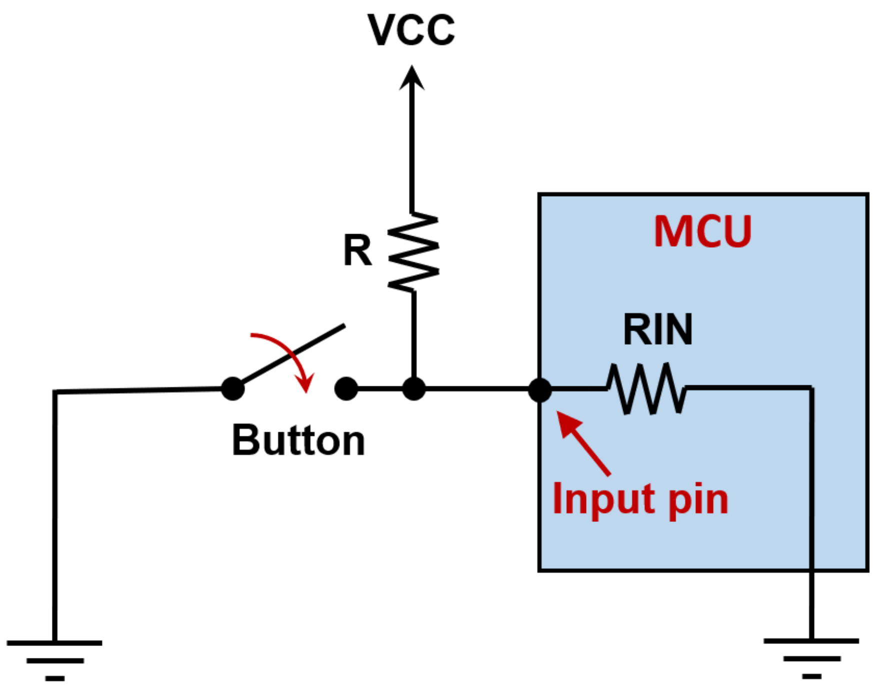
   Use the X-Y scatter plot in "`Landmark.xlsx`" to see whether you can deduce a few line segments based on this idea. Note that there is no need to actually perform the calculations (as excel is not very friendly ☹), rather use the idea above to see how "wall" segment can be found and plotted.

   Note: This question is really just to expose you to the idea. Don't panic as you wont be asked to implement the algorithm 😆.

   **Credit:** The idea of the questions and algorithm are from https://dspace.mit.edu/bitstream/handle/1721.1/119149/16-412j-spring-2005/contents/projects/1aslam_blas_repo.pdf

**Section 4. Power Management**

1. Pull-up Resistors




Pull-up (and also pull-down) resistors are very common when using microcontroller units (MCU). In earlier studios, you have used them together with push buttons to control the logic at input pins. Why are they needed? If nothing is connected to an input pin, its voltage will be floating, and the detected logic can go either high or low. To prevent this unknown state, we use these resistors to ensure that the pin is either in a high or low state, while using a small amount of current. In our studios, we always use R = 10 k∧. For a typical MCU input pin, its input resistance RIN needs to be high, and can be between 1 to 100 M∧. In this question, we try to understand whether R = 10 k∧ is a good choice. Assume VCC = 5 V, and RIN = 1 M∧.

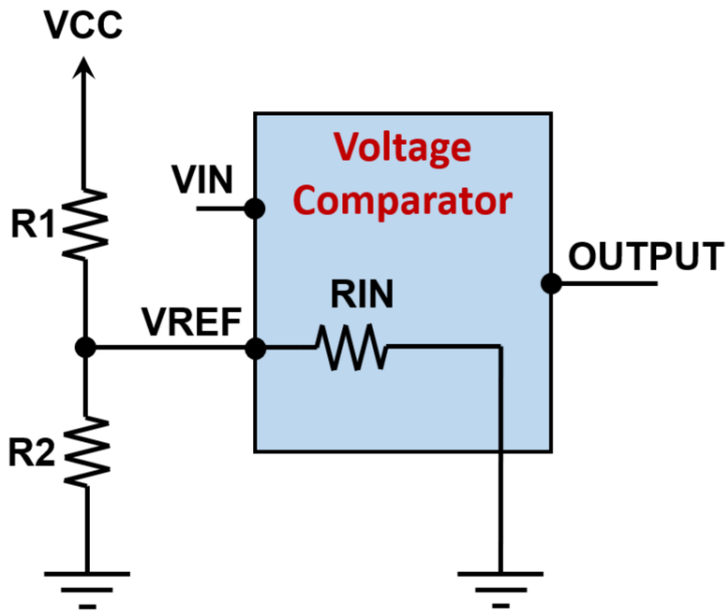
a. Can R be 0? Why not?

b. Assuming R = 10 k∧, what is the voltage at the input pin when the switch is opened, and what is the current drawn from the source VCC?

c. Repeat (b) when the switch is closed.

d. Suppose someone thinks that the current calculated in (c) is still too high, and decides to save power by using R = 1 M∧. what is the voltage at the input pin when the switch is open?

2. Using a Potential Divider to Provide Reference Voltage




Suppose you are working on an intelligent system that requires the use of a reference voltage for decision making. One possibility is to use a potential divider to obtain this reference voltage. Also suppose that low-power consumption is an important consideration in your design. If VCC = 5V, the required VREF is 2.5 V, and the input resistance RIN of the voltage comparator is 1 M∧, assess the suitability of each of the following choices of R1 and R2:

a. R1 = R2 = 100 ∧
b. R1 = R2 = 10 k∧
c. R1 = R2 = 1 M∧
d. R1 = R2 = 10 M∧