[[1]](#footnote-1)

C++ program for processing sensor data

*Name student Name Instructor*

*Nguyễn Hoàng Anh– 20192183 - ĐKTĐH- HTĐ TS. Hoàng Đức Chinh*

***Abstract*— The Internet of Things (IoT) describes the network of physical objects “things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools. With more than 7 billion connected IoT devices today, experts are expecting this number to grow to 10 billion by 2020 and 22 billion by 2025. However, the physical real-time IoT sensor data include several challenges, such as time, temperature, wind power, etc. As such, this paper addresses how to process IoT sensor data and convert it into binary packets. This paper addresses the data processing techniques such as converting data into the binary packet for cloud computing, fog computing and edge computing, towards various challenges in IoT sensor networks. In summary, this paper is to present a program of IoT sensor data processing.**

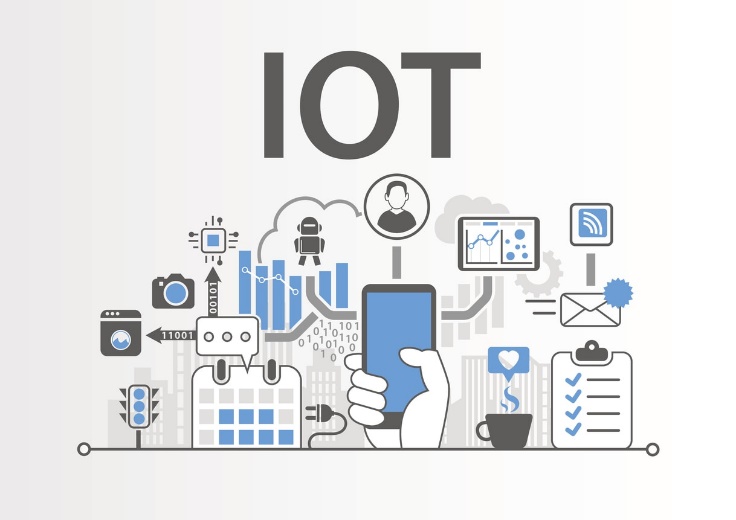
***Keywords*— sensing data, binary packet, sorting data, algorithms for binary processing.**

I. Induction

S

ensor data is the output of a device that detects and responds to some type of input from the physical environment such as temperature, humidity, etc. The output may be used to provide information or input to another system or to guide a process.

In this report, we show a program in C++ that opens a file and finds if it contains sensor data (in csv file). It will then convert the file to binary packet format and save the result in another file (in txt file). The program can handle any error that may occur by displaying a message with an error code in a log file, from which the user can identify the error code and check it out.



*Reality with IOT*

# II. The idea, design, and implementation of the program

* The idea

As following the project, the requirements are that: need to find a programming method which can be coded simultaneously and finely delineating work. Besides, need reusable function, inheritance of a function to other functions to compacting and simplifying. Finally, in order to optimize the project, need a good algorithm with most suitable time complexity which can deal with a large amount of data from file. So, coming up with ideas and used mainly these methods:

* **Three-layer architecture**
* **Data transfer object class**
* **Object-Oriented Programming (OOP)**
* **Hash table**
* **Linked list**
* **Algorithms: sort, transform format, …**

## A. Three-layer architecture

1. *What’s three-layer architecture?*

Layer indicates the logical separation of components. Layered architecture concentrates on grouping related functionality

within an application into distinct layers that are stacked vertically on top of each other. Each layer has unique namespaces and classes.



*Three–layer architecture*

1. *Information about three-layer follows:*
2. **Presentation layer**

It is the first and topmost layer present in the application where users can interact with the application.

1. **Business Logic layer**

This is the middle layer - the heart of the application. It contains all business logic of the application, describes how business objects interact with each other, where the Presentation layer and Data Access layer can indirectly communicate with each other.

1. **Data Access layer**

The Data Access layer is where Data is stored in the Database and get data from Database.

1. *Advantages of three-layer architecture*

• Explicit code: The code is separated into each layer. Each one is dedicated to a single responsibility such as interface, business processing, and querying instead of bundling all the code in one place.

• Easy to maintain: As its roles separate each layer, it would be easier to change something. The modification can be isolated in a single layer or affected

only the nearest layer without affecting the whole program.

• Easy to develop, reuse: When we want to modify a function, we can easily do that as we already have a standard architecture. In case we want to alter a complete layer such as from Winform to Web form, we only need to implement to replace the Presentation layer; other layers can be reused completely.

• Easy to transfer: We could save time on moving the application to others as they have a standard architecture to follow and apply.

• Easy to distribute the workloads: By organizing the code into different layers based on its responsibility, each team/member can write their code on each layer independently, which in turn helps developers control their workload.

1. *How does three-layer architecture work?*

In three-layer architecture, the Presentation layer doesn’t communicate directly with the Data Access layer. The Business Logic layer works as a bridge between the Presentation Layer and the Data Access layer. The three-layer architecture works as follows:

• The Presentation layer is the only class that is directly interacted with the user. It is mainly used for presenting/collecting data from users, then passes them to the Business Logic layer for further processing.

• After receiving information from the Presentation layer, the Business Logic layer does some business logic on the data. During this process, this layer may retrieve or update some data from the application database. However, this layer doesn’t take responsibility for accessing the database; it sends requests to the next layer, the Data Access layer.

• The Data Access layer receives requests from the Business Logic layer and builds some queries to the database to handle these requests. Once the execution gets done, it sends the result back to the Business Logic layer.

• The Business Logic layer gets responses from the Data Access Layer, then completes the process and sends the result to the Presentation Layer.

• The Presentation layer gets the responses and presents them to users via UI components.

* *An addition layer: DTO*

1. What is the Data transfer object class?

DTOs or Data Transfer Objects are objects that carry data between layers GUI, DAL, BLL in order to reduce the number of methods calls.

1. DTO layer structure in our project including

Fields: ID, Time, Temperature, Humility.

## B. Object-Oriented Programming (OOP)

1. *What is the OOP?*

Object-Oriented Programming (or OOP) is a paradigm of programming in which programs are written and structured around objects rather than functions or logic. Here, objects are defined as data fields that have unique attributes and behaviour. They contain data in the form of attributes and procedures in the form of methods. Object

procedures can access and modify the data present in an object.

1. *Why did I choose OOP for our project?*

We can build the programs from standard working modules that communicate with one another, rather than having to start writing the code from scratch which leads to saving of development time and higher productivity,

OOP language allows breaking the program into bit-sized problems that can be solved easily (one object at a time). OOP systems can be easily upgraded from small to large systems. It is possible that multiple instances of objects co-exist without any interference. It is very easy to partition the work into a project based on objects.

It is possible to map the objects in the problem domain to those in the program. By using inheritance, we can eliminate redundant code and extend the use of existing classes. Message passing techniques are used for communication between objects which makes the interface descriptions with external systems much simpler. The data-centered design approach enables us to capture more details of the model in an implementable form.

## C. Data structure

* *Hash table*

1. *What is the hash table?*

In computing, a hash table (hash map) is a data structure that implements an associative array abstract data type, a structure that can map keys to values. A hash table uses a hash function to compute an index.

1. *Applying to my project*

As a result of knowing all the binary code and hex code respectively, I try to build a perfect hash table for binary to hexadecimal base transformation.

Diagram

Description automatically generated

1. *Hash table for Transformation*

* First of all, writing a hash function and trying to find a perfect hashing. Because I know each binary code have a specific decimal value, so by using its attribute, my hash table is perfect.
* Second, we save hash values in an array with 16 elements.”.

**Table

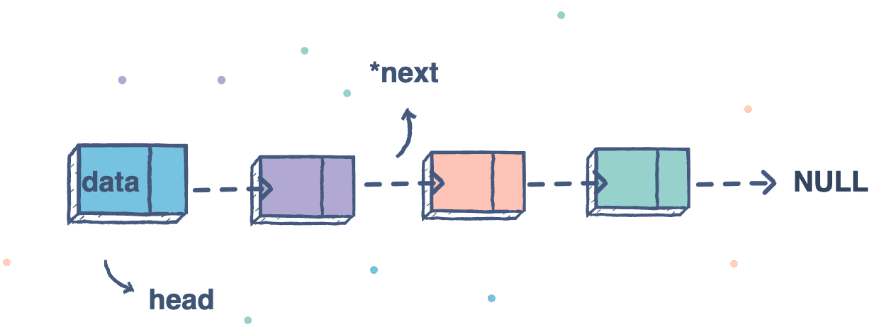
Description automatically generated with medium confidence**

*Format of BinaryToHexHashTable*

* *Linked list*

1. *What is the linked list?*

In computer science, a linked list is a linear collection of data elements whose order is not given by their physical placement in memory. Instead, each element points to the next. It is a data structure consisting of a collection of nodes that together represent a sequence. In its most basic form, each node contains data and a reference (in other words, a link) to the next node in the sequence. This structure allows for efficient insertion or removal of elements from any position in the sequence during iteration. More complex variants add additional links, allowing more efficient insertion or removal of nodes at arbitrary positions.



*Singly-linked list*

1. *Applying singly linked list to project*

A single-linked list is suitable for our project because we have to assign a fixed array to deal with getting data. The list can be increased its size automatically. It is suitable for storing data types such as Error, Morse Code and Text which we do not know the amount of

We use a standard library: the list to get the data structure list instead of writing ourselves a linked list because it is more optimized.

## D. The algorithm

* Sort Algorithm: advanced bubble sort

Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order. The pass of the list is repeated until the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller or larger elements "bubble" to the top of the list.

The time complexity of this algorithm is O(n^2). In the worst case, if the list is sorted, bubble sort will need a lot of time to run. Hence, I advanced the algorithm by using a bool variable (check) to make sure if no swapping exists, the bubble sort will be broken out.

The space complexity is O(1). And the bubble sort is a stable algorithm because if the list of sensor data is sorted by id order, and then sorted by time, all values that have a similar time value are also kept the order in id.

*Two main functions are presented by flowchart (on the final page).*

# III. Discussion and evaluation of the results

* *A few optional commands can be:*

o –m: print out the help to instruct the user of the usage on the screen and save the instruction to a file name readme.txt, override if readme.txt has already existed.

Text

Description automatically generated

* *A sample Input file (.csv)*

Text

Description automatically generated

o <inputfile> <outputfile>: forced to take the input file as a csv file and the output file as the text file. If no other argument is provided, the conversion is carried out by each line of the files.

Text

Description automatically generated

Text

Description automatically generated

o <inputfile> <outputfile> -s -id: forced to take the input file as a csv file and the output file as the text file. The data in the output file is sorted by id and in ascending order.

Text

Description automatically generated

Text

Description automatically generated

o <inputfile> <outputfile> -s -ti: forced to take the input file as a csv file and the output file as the text file. The data in the output file is sorted by time and in ascending order.

Text

Description automatically generated

Text

Description automatically generated

o <inputfile> <outputfile> -s -id -asc: forced to take the input file as a csv file and the output file as the text file. The data in the output file is sorted by id and in ascending order.

Text

Description automatically generated

Text

Description automatically generated

o <inputfile> <outputfile> -s -ti -asc: forced to take the input file as a csv file and the output file as the text file. The data in the output file is sorted by id and in ascending order.

Text

Description automatically generated

Text

Description automatically generated

o <inputfile> <outputfile> -s -id -des: forced to take the input file as a csv file and the output file as the text file. The data in the output file is sorted by id and in ascending order.

Text

Description automatically generated

Text

Description automatically generated

o <inputfile> <outputfile> -s -ti -des: forced to take the input file as a csv file and the output file as the text file. The data in the output file is sorted by id and in ascending order.

*Text

Description automatically generated*

*Text

Description automatically generated*

* *The program can handle any error that may occur by displaying a message with an error code from which the student can correct the lesson. Some cases are listed below.*

If the output file has existed, the output: Warning: D:\\output.txt already exists. Do you wish to overwrite (y, n)?

Text

Description automatically generated

(1)

The program cannot open the input file. message can be: “E007: FILENAME could not be opened”

A screenshot of a computer

Description automatically generated with medium confidence

(2)

If a command is missed or not correct form, the message can be:

*“Error E001: Unknown command. Type “SendingData -m” to help.*”

*“Error E002: Do not allow more than 6 arguments in the same command. Type "SendingData - m" for help.”,*

*“Error E003: Missing arguments. Type "SendingData - m" for help.".*

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

(3)

If each field from the sensor data is missed, an error will be created, all errors are indicated from the input file will be stored in a log file. If a line of input files did not have data, it will be throughout.

Text

Description automatically generated

(6)

* *What have been done and have not been done in our work:*

**We have completed all the cases required by the assignment.**

**In the hash table convert from morse code to text we have not tested 100% of the cases to avoid collisions.**

References

1. GeeksforGreeks, “List in C++ Standard Template Library (STL)”,5th Floor, A-118, Sector-136, Noida, Uttar Pradesh- 201305. [Online]. Available: <https://www.geeksforgeeks.org/list-cpp-stl/>
2. TOPDev, “Mô hình ba lớp (Three-layer)”, Viet Nam. [Online]. Available: <https://topdev.vn/blog/mo-hinh-3-lop-la-gi/>

Diagram

Description automatically generated

1. [↑](#footnote-ref-1)