

Efficient Scheduling System for a Company's Dispatch Trucks

M. O. Babatunde
Department of Electrical &
Computer Engineering
Ahmadu Bello University
Zaria, Nigeria.

E. A. Adedokun
Department of Electrical &
Computer Engineering
Ahmadu Bello University
Zaria, Nigeria.

Ajayi Ore-ofe
Department of Electrical &
Computer Engineering
Ahmadu Bello University
Zaria, Nigeria.

ABSTRACT

This paper presents the design of an efficient scheduling system for a Company's dispatch truck using Visual Basic as an object oriented programming language; MS SQL server as the database management system. Open Database Connectivity (ODBC) was used to connect the application to the database. SQL DML (Data Manipulation Language) expression has been used to query from the database. Stores in each state of federation have data centers connected to the central database at the factory. A photoelectric sensor is used to detect and count the number of trucks that enters the park at the factory. An update of the number of truck is regularly sent to the stores across the state to inform them about the number of truck remaining for the park to be filled.

Once the trucks deliver their products, they return to their various originating states. The database is updated as the number of trucks that has entered the park is reset. Messages are sent to the states whose trucks have not brought products to the factory, informing them of parking space in the factory. As product are delivered the entire process will be repeated.

The system ensured an efficient means for scheduling dispatch trucks, so that the required quantity of materials for production were always made available.

General Terms

Truck Scheduling, Vehicle Communication.

Keywords

Scheduling, Programming language, Manufacturing, Design, ODBC.

1. INTRODUCTION

Transport systems are the response to the ever-growing needs for contacts between individual and societies and for the movement of commodities as part of national and global economies. A factory needs regular supply of raw-materials for its production [8]. Most of these raw materials do not exist at the site of the factory, so the need arises to have dispatch trucks to move such raw-materials from their stores in various states of the federation to the factory.

The trucks will have a means by which they will be scheduled to transport materials to the factory. The whole scheduling process will be done with a software application that will have a database of all the trucks. The database will include the quantity of materials (in tons) that is required for production at the factory, the number of trucks that is required to meet such need. Once the required quantity is received, no other trucks are allowed into the factory. The software package will be used to schedule the number of trucks that comes into the factory based on the requirement of the factory.

2. METHODOLOGY

The methodology used in the development of the scheduling system involves Hardware and Software parts, discussed below:

2.1 Hardware

The Hardware part will be based on wireless networks of photoelectric sensors that will be deployed on the entrance of the factory's truck park. The sensor will monitor the number of trucks that goes in and out of the park. The sensor will also detect the passage of the trucks and send the information to the data center, where the software application will be. The application will be able to know the number of trucks that comes in and goes out of the park through the data sent from the sensor.

The sensor nodes send data packets that tells the application about the entry or exit of a truck out of the parking area. These packets are made up of:

- (i) identification number of the network and
- (ii) Identification number of the node.

The identification fields of the node and network will correspond in this case, to the store in any part of the country that the truck is coming from [1].

Two circuit boards were used for the hardware components of the nodes: a mainboard and a sensor board. The mainboard contains the components responsible for processing data and communication between nodes, while the sensor board is connected to the sensor. A box with IP67 protection houses the two circuit boards, the batteries and a solar panel, measuring 82 mm x 80 mm x 85 mm. The batteries are lithium polymer 1900 mAh, with a nominal voltage of 3.7 V, and the solar panel has a power output of 0.4 W (5 V/80 mA). The main components of the mainboard are the Intel 8051 with 8 KB of RAM and a programmable flash memory of 256 KB and the CC2530 system-on-chip from Texas Instruments [2] and a radio module in the integrated circuit. The radio module is a transceiver that operates in the 2.4 GHz band, compatible with the IEEE 802.15.4 protocol, with programmable output power of up to 4.5 dBm, allowing a maximum transmission speed of 250 Kbps. Other components include CC2591 amplifier, which also operates in the 2.4 GHz band. This amplifier extends the range of communications by providing a power amplifier that increases output power and improves receiver sensitivity.

Wireless communication is established using the 6LoWPAN standard that enables the use of IPv6 over the IEEE 802.15.4 standard. The main objective of using this standard is to be able to use IPv6 on a low-energy WSN, thereby enabling

small devices, with limited processing power, to achieve broad connectivity (interoperability) and to maximize the battery life [3].

The main board also has an RS-232 interface, JTAG, and a number of expansion connectors allowing communication

with external devices.

As shown in Figure 1, since the integrated circuits require 3V DC, a 3 V DC/DC converter has been added to the mainboard to regulate the 3.7 volts supplied by the battery to 3V required by the integrated circuits.

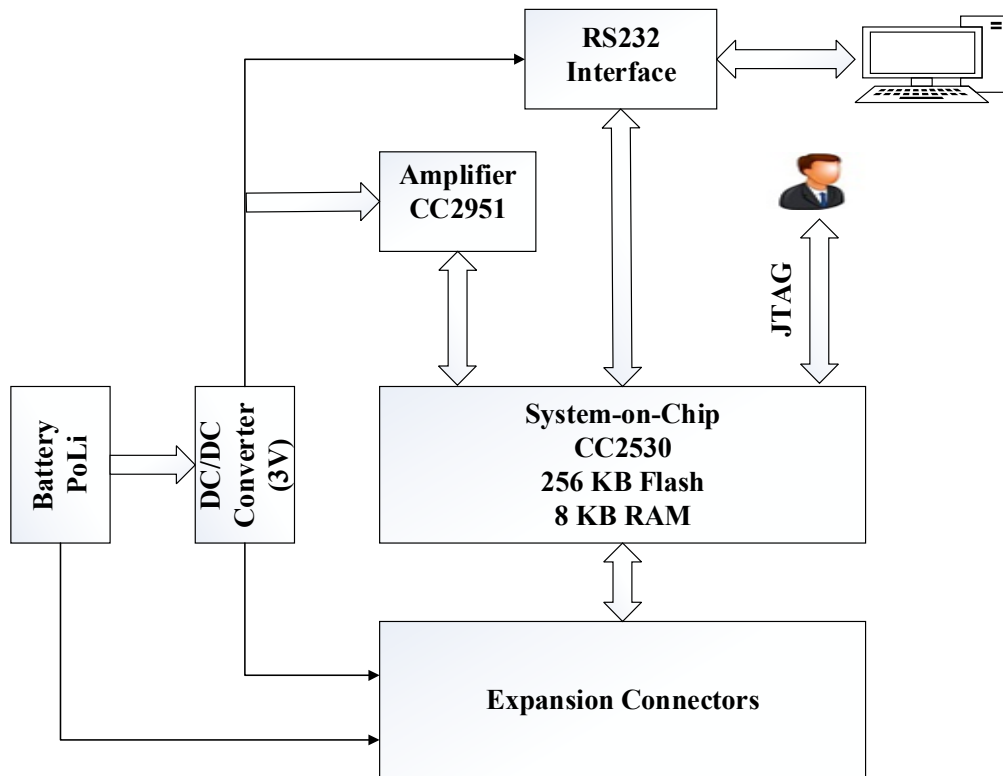


Figure 1: Block diagram of the mainboard (Juan et al., 2016)

As shown in Figure 2, the sensor board is made up of a series of voltage converters and interfaces to allow connection to the sensors. It has two DC/DC voltage converters, 12 V and 5 V, an SDI-12, and a 4–20 mA interface. This circuit board also features a battery charging system through an external solar panel, thus increasing the autonomy of the nodes.

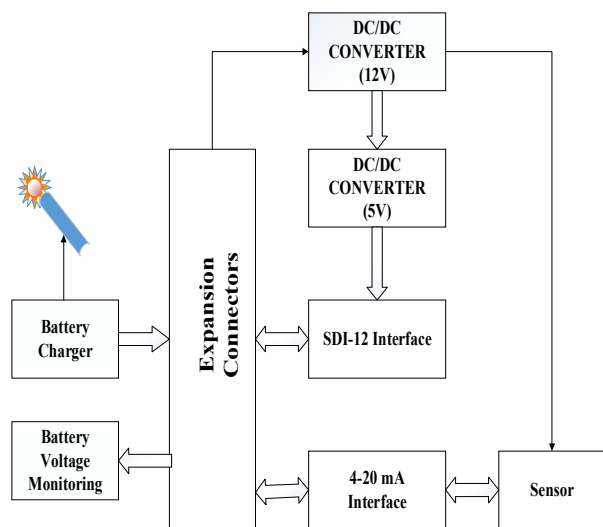


Figure 2: Block diagram of the Sensor board (Juan et al., 2016)

The photoelectric sensor used is a SHARP-GP2Y0D02YK0F (Sharp Corporation, Osaka, Japan) [4], shown in figure 3.



Figure 3: SHARP-GP2Y0D02YK0F sensor along with a standard AA battery (Juan et al., 2016)

The sensing mode to be used is the diffuse mode. The sensors will be deployed in speed bumps that will be built at the entrance of the truck park. Two sensors will be deployed on the speed bumps, so as to avoid a situation where a person or an object that is not a truck could be counted as a vehicle.

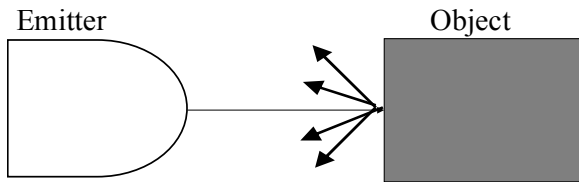


Figure 4. Diffuse Sensing Mode (Juan et al., 2016)

The Contiki Operating system and the library for CC2530 system-on-chip is used to develop the software for the sensor nodes. Contiki is an open-source operating system geared towards the Internet of Things. It enables communication with microcontrollers via the Internet and can run on a wide range of wireless low-power devices, providing mechanisms to estimate the total energy consumption of the system and to identify the units that consume the most. In addition, Contiki is designed for small systems with only a few kilobytes of memory available; it is very efficient at memory management and provides different memory allocation mechanisms. With regard to communications, Contiki is fully compatible with the IPv6 and IPv4 standards, providing a full IP network stack, with standard protocols such as User Datagram Protocol (UDP), Transmission Control Protocol (TCP), and Hypertext Transfer Protocol (HTTP). Furthermore, it also supports low-power wireless communications on the 6LoWPAN, Routing Protocol for Low-Power and Lossy Networks (RPL) and CoAP standards [5].

2.1.1 Data Gateway

The gateway node receives the packets sent by the sensor nodes (i.e. the number of trucks entering the park) and sends them on to the data centre. It also allows communications in the opposite direction (notifying the various stores if there is still parking space or not). From the hardware point of view, this element is composed of a mainboard similar to that used for the sensor nodes. Therefore, the gateway software will have characteristics and functionalities similar to that used for the sensor nodes without having to perform data packing or unpacking.

2.2 Software

The methodology used in the development of the scheduling system involve the use of Visual Basic to build the application with Microsoft SQL Server as the database RDBMS (Relative Database Management System) and Microsoft ODBC (Open Database Connectivity) to connect the application to the database.

The system administrator registers all the trucks that will be used to move materials from the site to factory. Some assumptions will be made in developing the system.

Assumption 1: - That the parking lot of the factory can only contain 100 trucks.

Assumption 2: - That the store in each state of the federation has 3 trucks each.

Assumption 3: - That 100 trucks of raw materials will be sufficient for production process to continue in the factory.

The K Means clustering algorithm is used by the trucks to find the best routes to the factory from their various states. The K means clustering algorithm is a partitioning algorithm used to divide the dataset into different parts. The objects are separated into k clusters, where k is a number which is known

before-hand. The dataset is partitioned into clusters such that each member of the dataset is assigned to some cluster. The concept of center of gravity, is introduced which is the arithmetic mean of the objects in that cluster. Each point is assigned to the closest cluster based on the defined distance measure. To measure the distance between the object and the center of gravity, we can use the Euclidean Distance Formula. [6 - 7]

To calculate Euclidean distance, let vectors $x=(x_1,x_2,\dots,x_n)$ and $y=(y_1,y_2,\dots,y_j)$. The formula for multi-dimensional distance [9] is given below:

$$d(x,y)= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (1)$$

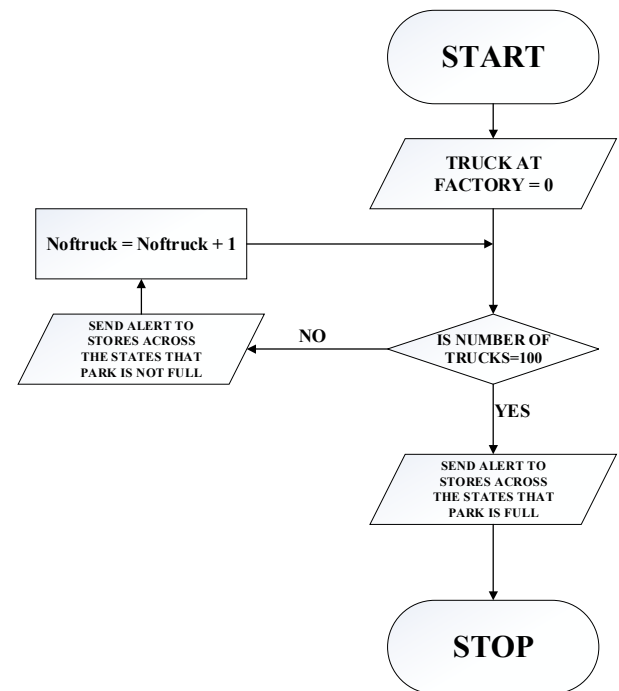


Figure 5: Flowchart for truck scheduling system

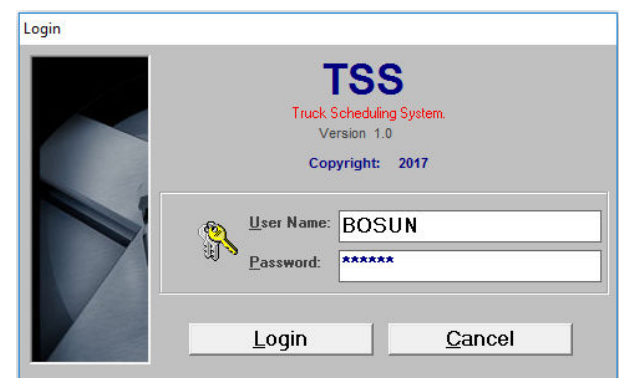


Figure 6: Login page

As shown in figure 6, the login page is used to register all the users of the application at the various data centers (the factory and the stores across the country). A unique user name and password is required for this system.

Table 1. Design view of table ‘users’

| Column Name | Data Type | Length | Allow Nulls |
|-------------|-----------|--------|-------------|
| sn | int | 4 | ✓ |
| username | nvarchar | 50 | ✓ |
| password | nvarchar | 50 | ✓ |
| secCode | Nvarchar | 50 | ✓ |
| Active | char | 2 | ✓ |

Table 2. Table ‘users’ with sample data

| sn | username | password | secCode | Active |
|----|----------|-----------|---------|--------|
| 1 | BOSUN | balama | 1781 | Y |
| 2 | BALA | kay | 74832 | Y |
| 3 | MOHAMMED | Haruna@01 | 37252 | Y |

Tables 1 and 2 show the design and data views respectively of table ‘users’.

Once a user logs in with a correct credential, the various truck of the company can now be registered, as shown below:

Figure 7: Truck registration page

All the various trucks across the stores nationwide are registered. All the data will be stored in a central data center at the factory.

Table 3. Design view of ‘Truckreg’ table

| Column Name | Data Type | Length | Allow Nulls |
|-------------|-----------|--------|-------------|
| sn | int | 4 | ✓ |
| TruckID | nvarchar | 10 | ✓ |
| Truckname | nvarchar | 50 | ✓ |
| PlateNumber | nvarchar | 10 | ✓ |
| Colour | char | 20 | ✓ |
| DriverName | nvarchar | 50 | ✓ |

Table 4. Table ‘Truckreg’ with sample data

| sn | Truck ID | Truck name | Plate Number | Colour | Driver Name |
|----|----------|----------------------|--------------|--------|----------------------|
| 1 | TR001 | FORD | GGE768 WA | Black | Danju ma Moh’d |
| 2 | TR002 | MERCE DES BENZ | ABJ678 XX | Red | James John |

Figure 8: Truck Dispatch form

Figure 8 shows the main dispatch form for trucks across the various stores. As the trucks are dispatched, an update is made on the database that logs the particular truck as coming to the factory, from the respective store. The same is done for trucks coming from various stores across the country. If all conditions remain favorable, and the truck gets to the factory, and crosses the sensors, it is counted and the counter is incremented until the truck gets to its maximum number (100), and no more trucks are scheduled to come to the factory from the stores.

Figure 9: Truck park update form

Figure 9 shows interface form that the sensor communicates with. As the trucks crosses, the sensor, it increments the ‘truck at the park’ textbox and displays the remaining amount of trucks the park can accommodate.

The data centre will receive information transmitted by the gateway and from the sensor nodes of the network, indicating the passage of trucks. It will also send the relevant instructions to the sensor network and store system data.

The number read from the passage of the trucks is stored in the software application in the server. The application increments the value, online, real-time, as it receives it from the sensor node. The system sends alert of space availability to the various data centers of the factory’s stores across the country, and also tells them the park is full when it receives signal from the sensor that the number of trucks at the park is 100 (one hundred).

Table 5. Design view of ‘Truckmgr’ table

| Column Name | Data Type | Length | Allow Nulls |
|-------------|-----------|--------|-------------|
| sn | Int | 4 | ✓ |
| Arrdate | datetime | 8 | ✓ |
| TruckRegNo | nvarchar | 10 | ✓ |
| TPark | int | 4 | ✓ |
| RemTPark | int | 4 | ✓ |
| SendAlert | char | 2 | ✓ |

Table 6. Table ‘Truckmgr’ with sample data

| sn | Arr date | Truck Reg No | TPark | RemT Park | Send Alert |
|----|-----------|--------------|-------|-----------|------------|
| 1 | 3/23/2017 | JJJ68AW | 1 | 99 | Y |
| 2 | 3/23/2017 | PKA789QQ | 2 | 98 | Y |
| 3 | 3/23/2017 | AAA432PA | 3 | 97 | Y |
| 4 | 3/23/2017 | ZAR324RE | 4 | 96 | Y |
| 5 | 3/23/2017 | GGE768WA | 5 | 95 | Y |

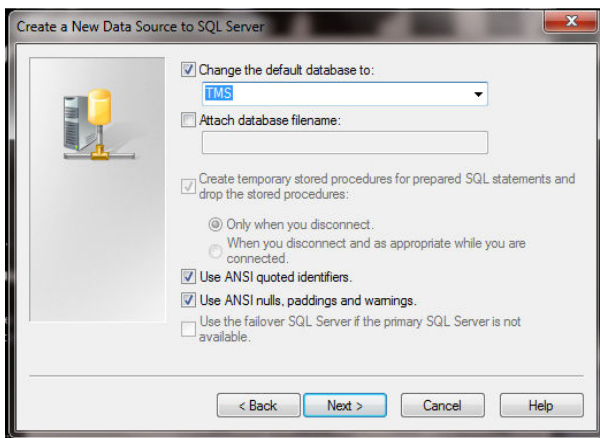


Figure 10: Data source connection

The Open Database Connectivity (ODBC), which is an application programming interface for accessing a database, is used to connect the Visual Basic application to the SQL server database. The connection code, “ con.Open "dsn=tms,uid=sa" “, is also added in the code of the various forms that makes up

the application.

2.3 Future Work

For this project, the technologies used are as follows: Visual Basic, MS SQL Server. It is suggested that the use of other object oriented programming language such as C#, Java and other tools such as HTML, JavaScript, Razor CSS and Relational database management system such as MySQL and Oracle should be explored.

3. CONCLUSION

This paper presented the design of an efficient scheduling system for a Company’s dispatch truck using Visual Basic programming language. The hardware and software part of the design were made and properly integrated. The system achieved a significant improvement in the scheduling of trucks for conveyance of raw materials from the stores to the factory.

4. REFERENCES

- [1] A. Q.-A. Juan A. Vera-Gómez, Carmelo R. García, Raúl Suárez Moreno, Fernando Guerra Hernández, : An Intelligent Parking Management System for Urban Areas. Sensors 2016 vol. 16: pp 4-8,.
- [2] "Texas Instruments Incorporated, CC2530 (Rev. B)," Available online: <http://www.ti.com/product/CC2530/technicaldocuments> (accessed on 23 April 2017).
- [3] M. Geoff, "The 6LoWPAN architecture," In Proceedings of the 4th Workshop on Embedded Networked Sensors, EmNets '07, Cork, Ireland, pp. 25–26 June 2007.
- [4] "SHARP Corporation, GP2Y0D02YK0F," Available online: http://www.sharp.co.jp/products/device/doc/opto/gp2y0d02yk_e.pdf (accessed on 23 April 2017).
- [5] "Contiki-Developers, Contiki 2.6.," Available online: <http://contiki.sourceforge.net/docs/2.6/> (accessed on 23 April 2017).
- [6] H. L. Jie Tang, Yunbo Cao, and Zhaohui Tang, (2005) : Email data cleaning : In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining KDD '05, vol. ACM, New York, NY, USA, , pp. 489-498.
- [7] P. J. R. Leonard Koufman, Vojtech Juhász (2012): Clustering by means of medoids: Full-text search in email archives using social evaluation, attached and linked resource.
- [8] R. Tolley, B. J. Turton, "Transport systems, policy and planning: a geographical approach", 2014.
- [9] "Smart Task Scheduling for a Truck Company" Available online: <http://library.ndsu.edu/.../Smart%20Task%20Scheduling%20for%20a%20Truck%20Company.pdf> (accessed on 23 April 2017).