Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Especialidad en Sistemas Embebidos

An Embedded System Design for Data Collection and Analysis for Smart Fitness Centers

---

TRABAJO RECEPCIONAL que para obtener el **DIPLOMA** de **ESPECIALISTA EN SISTEMAS EMBEBIDOS**

Presenta: **ERICK EDUARDO VITE LOPEZ**
JAIME LEONEL NAVARRO OCAMPO

Asesor **LUIS ENRIQUE GARABITO SIORDIA**

Tlaquepaque, Jalisco. 2 de diciembre de 2021.

An Embedded System Design for Data Collection and Analysis for Smart Fitness Centers

Jaime Leonel Navarro Ocampo
Department of Electronics, Systems and Informatics
Instituto Tecnológico de Estudios Superiores de Occidente
Guadalajara, Mexico
jaime.navarro@iteso.mx

Erick Eduardo Vite Lopez
Department of Electronics, Systems and Informatics
Instituto Tecnológico de Estudios Superiores de Occidente
Guadalajara, Mexico
erick.vite@iteso.mx

Luis Enrique Garabito Siordia
Department of Electronics, Systems and Informatics
Instituto Tecnológico de Estudios Superiores de Occidente
Guadalajara, Mexico
garabitosiordia@iteso.mx

***Abstract*—The application of embedded systems in fitness centers is limited. Therefore, this paper proposes a system that can be integrated into weight machines to incorporate the benefits of the Internet of Things by collecting exercise routine data. A prototype consisting of a FRDM-K64 development board, NFC and distance sensors is proposed. Data of exercise routines such as repetition distance and time intervals were successfully measured by the prototype. Future applications include expanding this concept to cardiovascular or resistance equipment.**

Keywords—IoT, embedded system, data analysis, weight machines, fitness centers

I. Introduction

The Internet of Things (IoT) has been defined as a world of interconnected things that collect and process information to perform specific tasks [1]. Due to its wide user acceptance in recent years, the rise of the IoT technologies has expanded from vending machines [2][3] to home automation [4], including process interconnection for industries, food monitoring for farms, and fitness centers, among others. Particularly, the application of IoT technologies in fitness equipment, is limited. Current approaches of these technologies for fitness centers include theoretical frameworks [5], however commercial applications are scarce.

Currently, fitness centers that call themselves "smart gyms" only use technologies consisting of screens as a visual aid to explain how the fitness equipment is used, and simple buttons that the user can press to request assistance from a coach. Furthermore, one application of IoT technologies is an intelligent dumbbell based on acceleration sensors that collect motion information, which is then transmitted through wireless communication technology to a mobile phone after being processed by a microcontroller. The user can then consult this information to make decisions about different fitness modes to achieve specific goals [6].

Based on the above, current weight machines do not take advantage of IoT technologies, such as data processing, cloud integration and user experience. Therefore, this paper proposes an embedded system that aims to integrate IoT technologies with weight machines to collect and process data from exercise routines to improve overall user performance, such as incorrect repetition intervals, speed and possible weight adjustments to improve overall performance.

This paper is organized as follows: Section II presents an overview of the architecture, hardware used, software development and prototype construction. Section III presents the obtained results, conclusions and future work.

II. Prototype Development

For this project, a modeling and architectural design methodology known as COMET/RTE (Concurrent Object Modeling and Architectural Design Method for Real-Time Embedded Systems) is used as it suits real-time embedded systems. The benefits of this methodology come from the highly iterative use case driven and object-oriented focus that addresses the requirements, analysis, and design modeling phases of the system and software development life cycle [7].

Fig. 1 shows the different steps in the COMET/RTE development life cycle model for embedded software. Following this methodology, each step is described in more detail below.

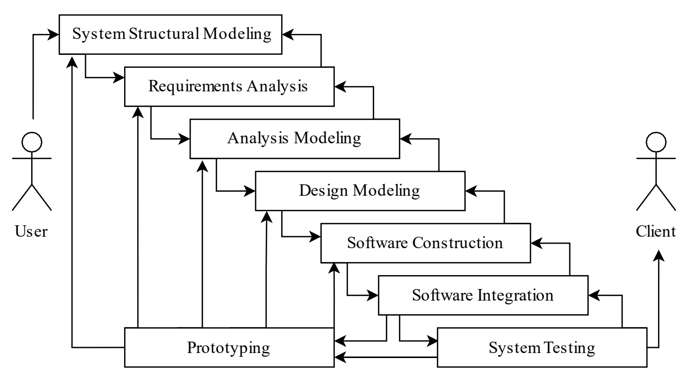


Fig. 1. COMET/RTE software development life cycle model.

A. System Structural Modeling

From a total system perspective, the problem domain for an embedded system implementation in a fitness center includes the following elements:

- *Embedded system for data collection and processing*, which is the main entity of the system
- *Weight Machines*, as the physical entities that are monitored by the system.
- *Database*, which is an external system where the information is uploaded.

- *User*, who controls the weight machines and observes the embedded system.

Fig. 2 proposes the conceptual structural model of the proposed problem domain.

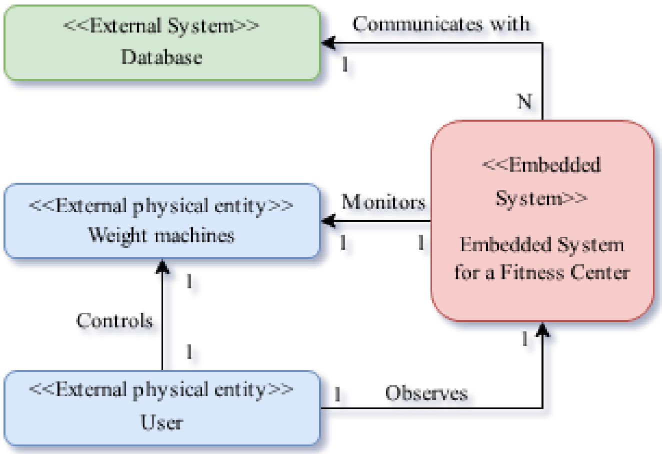


Fig. 2. Model of the total system perspective.

B. Requirements Analysis

The requirements that the prototype must fulfill are the following:

- System shall be able to identify who is using the gym equipment.
- System shall be able to monitor and measure exercise routine data in real time.
  - System shall be able to monitor and measure the weight been used by the gym equipment.
  - System shall be able to monitor and measure the number of repetitions performed by the user.
  - System shall be able to monitor and measure the time it took the user to perform each repetition.
- System shall have an internet connection
- System shall be able to process the exercise routine data and then send it to a database.
- The device must be compact enough to be installed in weight machines. The maximum dimensions for any module are the following:
  - 26mmx64mmx90mm
- The device must be modular to be able to replace its components easily.

C. Analysis and Design Modeling

A layered architecture consisting of two main layers is proposed for this prototype (Fig. 3):

1. Software design, which contains the operating system and the different software modules, as well as the database where the data will be uploaded.
2. Hardware design, which includes the development board and sensors that are interconnected in a modular way.

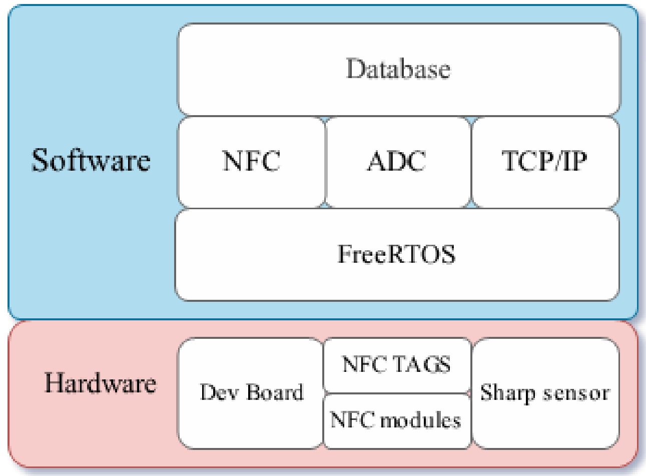


Fig. 3. Overview of the prototype layered structure and layer contents.

1) Hardware Design

The hardware selected for the prototype consists of two NFC (Near-Field Communication) modules, where one is used to read the contents of NFC tags that will be installed in the weight machines and the other is used to read the NFC user tag, which contains the user information. An infrared optical sensor will handle distance measuring, and an SD card will function as local storage if the system is not able to access the online database. All the components will be connected to a development board, which will process the data collected by the sensors and send it to a database.

The relationship between these hardware components is shown in Fig. 4

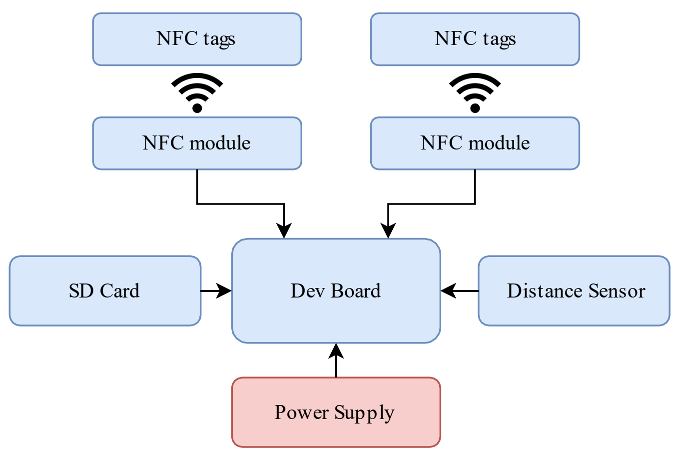


Fig. 4. Overview of the hardware components and their relationships.

The prototype is intended to be installed in integrated weight machines, so it is especially important that the overall size of the system does not exceed the required dimensions. The development board that was selected for this prototype is the FRDM-K64 from NXP, since it offers a balance between flexibility for efficient prototyping and processing power. To complement this development board, the OM5578 NFC reader was selected due its compatibility with the board header layout, allowing for plug and play integration. As for the distance sensor, the infrared Sharp GP2Y0A21 was selected due to its measuring range. Table 1 shows the characteristics of the selected hardware.

TABLE I. PROTOTYPE HARDWARE DESCRIPTION

Item	Hardware	
	Component	**Description**
1	FRDM-K64F	- ARM® Cortex™-M4 32-bit core with DSP - 120 MHz max CPU frequency - 256 KB RAM - 1 MB flash memory - Kinetis K64 USB - K6x Ethernet
2	NFC module OM5578	- Supply voltage: 3.3 V, 5 V optional - Current limit threshold 180 mA - Integrated high performance RF antenna
3	NFC tags NTAG216	- 888 bytes of memory - Waterproof - Password-protectable
4	SD Card	- FAT file system - Operating Voltage: 2.7V to 3.6V
5	Sharp GP2Y0A21	- Operating voltage: 4.5 V to 5.5 V - Average current consumption: 30 mA - Measuring range: 10 cm to 80 cm - Output: Analog voltage from 1 V to 3.3 V
6	Power Supply	- Voltage range: 1.71 to 3.6 V - Max. current consumption: 25mA

2) *Software Design*

The proposed real-time software architecture for the system consists of an implementation of the FreeRTOS operating system for embedded devices, where tasks for the NFC and ADC peripheral software, and the TCP/IP module for the database communication are implemented. The sublayers of the software layer are described below.

a) **FreeRTOS***:* The first sublayer consists of the operating system for the microcontroller. FreeRTOS offers robust task scheduling and system resource management which are neccesary for real time applications where tasks must be performed within specified time constraints. This operating system is designed for embedded applications where memory is often limited, so it comes with just the neccesary functionality for real time scheduling, such as inter-task commuication and timing including synchronisation mechanisms. It also offers the possibility to enable additional functionality by using add-on modules, such as networking stacks, for example [8].

b) **NFC, ADC and TCP/IP***:* The second sublayer consists of the different modules for the hardware peripherals. This includes the module for the NFC readers, which offers the functionality of reading NFC tag contents. The ADC module for the SHARP laser sensor is used to measure distance, which is necessary because the sensor is analog. The TCP/IP module provides access to the internet where communication to an external database can be enabled to upload the data measured by the sensors.

c) **Database:** The third sublayer consists of the communication between the embedded system and a database where the measured data is uploaded and stored using PHP requests that send the information to a relational database that will be subject of analysis, presentation, and reporting.

D. *Software Construction and Integration*

The developed code consists of an implementation of FreeRTOS where different tasks that are run in set time intervals by a priority based preemptive task scheduler are created. Five tasks are then defined:

1. System initialization: contains the different initialization routines of the different modules, such as the NFC, ADC module. The connection to the database is also established.
2. User login: handles the credentials of the system user. This task interacts with the NFC module to read the information of NFC tags where the user identity code is stored, allowing login functionality. This task also interacts with the database task to verify the user's credentials.
3. Database: uploads the measured data related to each user to an external database.
4. Process data: uses the NFC and ADC modules to monitor and collect data from the user's exercise routines. This task interacts with the database task to upload the user's exercise data.
5. USB: handles local data storage if the online database is not available. If the database comes back online, this task will upload the stored routines in the SD card memory.

The way in which these previously mentioned tasks will be interacting with each other is defined in the following sequence diagram shown in Fig. 5.

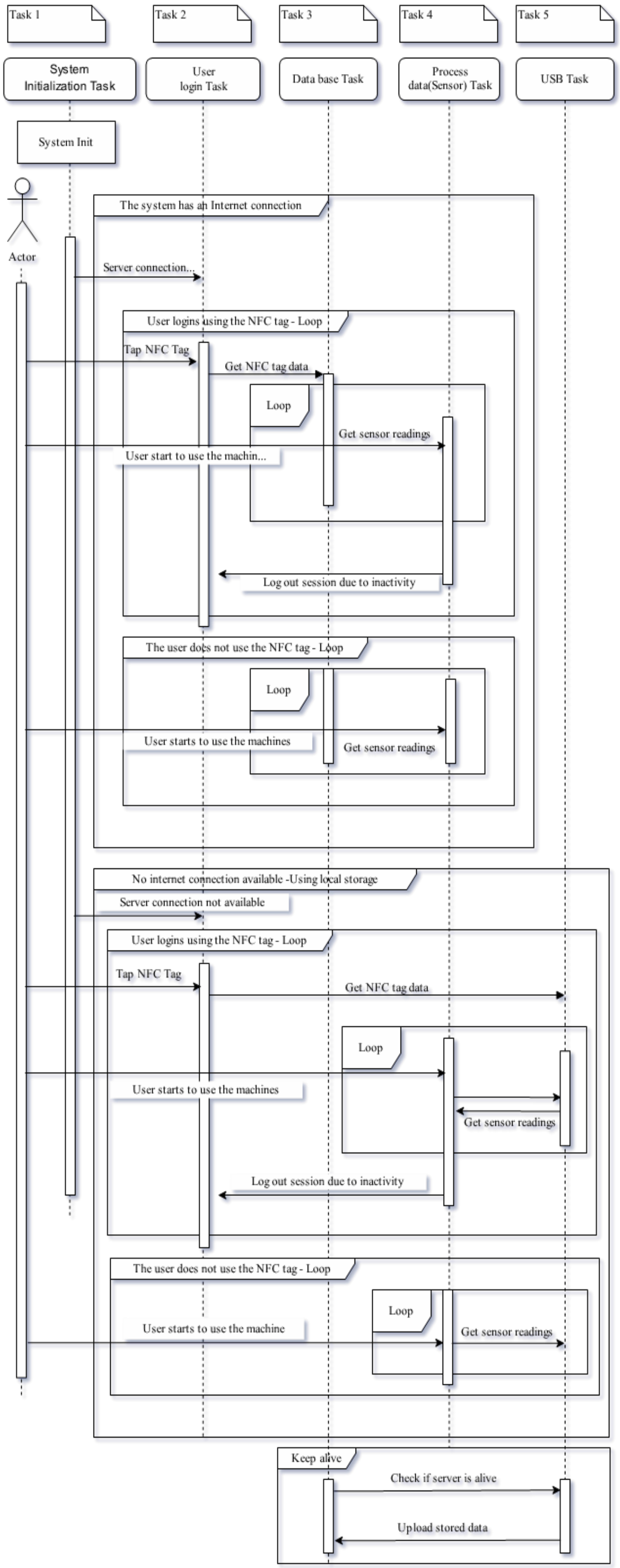


Fig. 5. Task sequence diagram where the interaction between tasks is illustrated.

E. Prototype Construction

The prototype is intended to be installed in integrated weight machines, so it must be compact and discrete so that it does not interfere with the functionality of the machine or the user. The prototype construction begins with the FRDM-K64 microcontroller where the SHARP distance sensor and the NFC modules are connected. As for the network connectivity, a regular ethernet cable is connected directly to the RJ45 port of the development board, enabling internet access for the database. The assembled prototype is shown in Fig. 6.

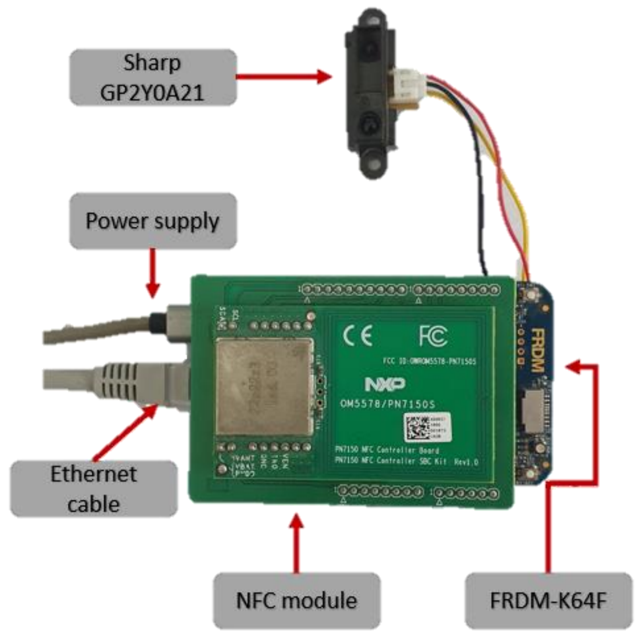


Fig. 6. Prototype implementation.

III. RESULTS

The assembled prototype can read the contents of a user identification card which contains the NFC tag. The contents are then uploaded to a database where a record of the user's login data is kept, including the timestamp, the weight machine that is being used and its location as shown in Fig. 7.

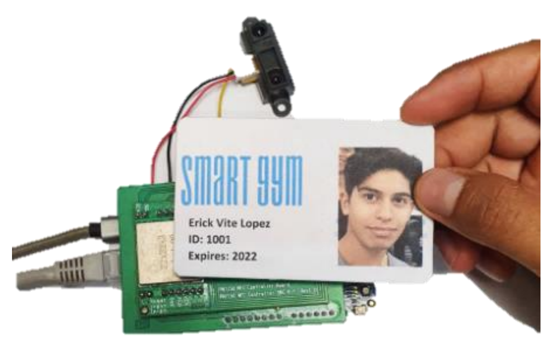
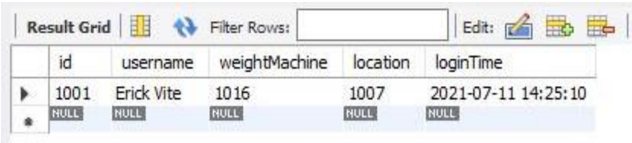


Fig. 7. Prototype reading an NFC card.

The database holds the login information, which can be later exported and analyzed to keep track of the multiple users of an integrated weight machine as shown in Fig. 8.



	id	username	weightMachine	location	loginTime
▶	1001	Erick Vite	1016	1007	2021-07-11 14:25:10
*	NULL	NULL	NULL	NULL	NULL

Fig. 8. Database with the user information.

The prototype can also measure the user's repetition distance using the SHARP sensor, where the iterations are uploaded to the database. This data can then be exported and plotted to analyze if the distance range is ideal depending on the user's specific goals. An example of this can be seen in Fig. 9, where a goal of 30 centimeters is set by the user for each repetition. After analyzing the iteration graph that shows the distance of the repetitions over time and the set goal, the user realizes the workout was not as efficient considering some of the repetitions did not meet the distance goal.

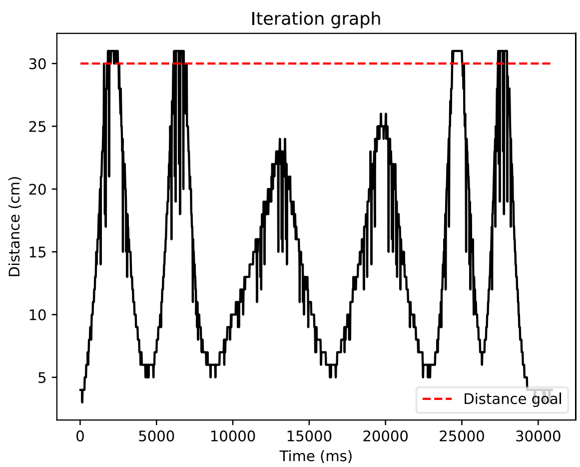


Fig. 9. Analysis of measured distance data.

Users can algo set time goals. Using the previous example, the user can also set a repetition time interval goal of 5 seconds. Looking at the time axis in Fig. 9, some the iterations were not performed within 5 seconds, so the goal was not always achieved.

CONCLUSIONS AND FUTURE WORKS

The prototype can identify the user of an integrated weight machine and collect exercise routine data, such as repetition distance and time intervals. This data is uploaded to a database where it can be exported and analyzed. However, this concept is currently limited to integrated weight machines, so the intention for future works is to adapt this concept to different fitness center machines to create an ecosystem where all the machines are connected with a common system.

ACKNOWLEDGMENT

This work was supported by the Mexican federal government through the CONACyT agency.

REFERENCES

[1] Qusay F. Hassan, "Introduction to the Internet of Things," in Internet of Things A to Z: Technologies and Applications , IEEE, 2018, pp.1-50, doi: 10.1002/9781119456735.ch1.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] "The "Only" Coke Machine on the Internet". Carnegie Mellon University. Retrieved 10 November 2014.

[3] Teicher, J., 2018. The little-known story of the first IoT device. [Online] IBM Industrious. Available at: https://www.ibm.com/blogs/industries/little-known-story-first-iot-device, accessed on Sept. 20, 2021

[4] V. L. K. B. Manda, V. Kushal and N. Ramasubramanian, "An Elegant Home Automation System Using GSM and ARM-Based Architecture," in IEEE Potentials, vol. 37, no. 5, pp. 43-48, Sept.-Oct. 2018, doi: 10.1109/MPOT.2016.2515644.

[5] A. Jain, "A Smart Gym Framework: Theoretical Approach," 2015 IEEE International Symposium on Nanoelectronic and Information Systems, 2015, pp. 191-196, doi: 10.1109/iNIS.2015.32.

[6] Y. Liu, C. Hsu and Y. Liu, "An Intelligent Dumbbell Designed for Fitness Activity Monitoring," 2019 IEEE International Conference on Computation, Communication and Engineering (ICCCE), 2019, pp. 29-32, doi: 10.1109/ICCCE48422.2019.9010796.

[7] H. Gomaa, "Overview of Real-Time Software Design Method for Embedded Systems," in Real-Time Software Design for Embedded Systems, Cambridge: Cambridge University Press, 2016, pp. 51–60, doi: 10.1017/CBO9781139644532.005.

[8] FreeRTOS. 2021. FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions. [online] Available at: https://www.freertos.org/index.html Accessed 18 November 2021.