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

Layered Software Architecture for Nanosatellites

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

Presenta:
DANIEL DIAZ PLATON
MAURICIO AMADOR CARRILLO
LUIS FLORES CAMPILLO
LUIS ÁNGEL HERNÁNDEZ GARCÍA
BRAULIO RAFAEL ROMERO NAVARRO
JOSE ALBERTO RIVAS SAUCEDA

Tutor **LUIS RIZO DOMÍNGUEZ**

Tlaquepaque, Jalisco. agosto de 2023

Layered Software Architecture for Nanosatellites

Mauricio Amador Carrillo
Specialization in Embedded Systems
ITESO Instituto Tecnológico de Estudios Superiores de Occidente
Tlaquepaque, Mexico
ie688046@iteso.mx

Daniel Díaz Platón
Specialization in Embedded Systems
ITESO Instituto Tecnológico de Estudios Superiores de Occidente
Tlaquepaque, Mexico
daniel.diazp@iteso.mx

Luis Flores Campillo
Specialization in Embedded Systems
ITESO Instituto Tecnológico de Estudios Superiores de Occidente
Tlaquepaque, Mexico
luis.flores@iteso.mx

Luis Ángel Hernández García
Specialization in Embedded Systems
ITESO Instituto Tecnológico de Estudios Superiores de Occidente
Tlaquepaque, Mexico
luis.hernandezg@iteso.mx

Jose Alberto Rivas Sauceda
Specialization in Embedded Systems
ITESO Instituto Tecnológico de Estudios Superiores de Occidente
Tlaquepaque, Mexico
alberto.rivas@iteso.mx

Braulio Rafael Romero Navarro
Specialization in Embedded Systems
ITESO Instituto Tecnológico de Estudios Superiores de Occidente
Tlaquepaque, Mexico
braulio.romero@iteso.mx

Dr. Luis Rizo Domínguez
Specialization in Embedded Systems
ITESO Instituto Tecnológico de Estudios Superiores de Occidente
Tlaquepaque, Mexico
lrizo@iteso.mx

Abstract **— Layered software architecture has been used to increase code portability and reduce development times during the Systems Development Life Cycle (SDLC). The main objective of the EMIDSS-4 project is to develop a reliable and scalable hardware and software layered architecture for data acquisition in nanosatellites. An iterative hardware/software co-design approach was employed, transitioning from an FPGA architecture to an NXP-based microcontroller. The results demonstrate that the new architecture provides continuous and adaptable data acquisition capabilities, enhancing the long-term viability of the project. In conclusion, this study contributes to the field of space exploration and lays the foundation for future research in nanosatellites.**

Keywords — Nanosatellite, Microcontroller, FPGA, PCB, Sensor, Software/Hardware architecture, AUTOSAR.

I. INTRODUCTION

Over the years, the accessibility to innovation and technology has enabled universities to invest in space exploration. Nanosatellites aside from a lower weight and cost are more affordable to launch into orbit and have a shorter development cycle compared to conventional satellites [1]. A main advantage of nanosatellites is that they are launched in low circular or elliptical orbits (with altitudes between 400 and 650 km) and travel at a speed of 8 km per second. By orbiting closer to the Earth, they not only guarantee optimum conditions for land observation or communications but are also better protected from solar and cosmic radiation [2]. The Experimental Module for the Iterative Design for Satellite Subsystems (EMIDSS), a collaboration between the National Autonomous University of Mexico (UNAM), the National Polytechnic Institute (IPN) and the Western Institute of Technology and Higher Education (ITESO) focuses on the remote sensing of the environment as part of NASA´s Mission FY 23 Balloon Campaign.

EMIDSS has undergone three previous versions prior to its latest edition, EMIDSS-4. The third version, EMIDSS-3, employed a Field Programmable Gate Array (FPGA) architecture, utilizing an iterative hardware/software co-design approach to create a nanosatellite [3]. This implementation successfully addressed the challenges of data acquisition, allowing an efficient capture and processing of diverse variables such as temperature, pressure, moisture, and air quality. However, it became evident that the FPGA-based solution lacked scalability over time, hindering its long-term viability due to the lack of documentation, which made it difficult to implement and adapt for future requirements. Consequently, the EMIDSS-4 project aims to develop a reliable and scalable software and hardware architecture by transitioning from FPGA to a microcontroller based on NXP architecture [4] to promote continuous and adaptable data acquisition capabilities.

This document is organized as follows: Section II outlines the proposed system design and specifics sensor integration. Section III discusses the implications of these results, draws conclusions about the potential applicability of the Printed Circuit Board (PCB) in future satellite missions, and outlines the future trajectory of the research. Section IV contains the overall concluding remarks.

II. PROPOSED SYSTEM DESIGN

A) HARDWARE ARCHITECTURE

The hardware architecture of the EMIDSS-4 satellite project is based on the following components:

- Microcontroller Selection

The EMIDSS-4 project utilizes the FSM32k144 microcontroller manufactured by NXP as the central processing unit for the satellite's hardware. This microcontroller features an ARM Cortex-M4 core, providing the necessary computational capabilities and peripheral integration required for effective satellite operations.

- Sensor Integration

To sense the environment a diverse set of sensors were implemented. The integrated sensors are described below:

- ➢ SGP40-D-R4 Air Quality Sensor: Enables accurate measurement of air quality parameters, particularly volatile organic compounds (VOCs) including CO2 (ppm) readings and total volatile organic compounds (TVOC) [ppb], contributing to a comprehensive understanding of the environmental conditions [5].
- ➢ MS860702BA01-50 Humidity and Temperature Sensor: Provides precise readings of humidity air percentage and temperature (Celsius degrees) levels, facilitating analysis of environmental changes and their impact on various ecosystems [6].
- ➢ MS5803-14BA Pressure Sensor: Allows accurate measurement of atmospheric pressure (Millibars). This data is valuable for applications such as weather monitoring and altitude determination [7].
- ➢ FSM300 IMU Sensor: Incorporates accelerometers, gyroscopes, and magnetometers to capture motion and orientation data represented in Euler vector. This sensor integration enhances the satellite capabilities, enabling applications such as attitude control and orientation tracking [8].

- Data Storage and Communication

To ensure efficient data storage and communication, the hardware architecture incorporates the SST26VF032B-104V/SM SPI memory module. This module provides ample storage capacity and high-speed data transfer capabilities, enabling reliable storage of sensor readings. Additionally, the architecture incorporates communication modules compatible with the PC104 protocol, allowing seamless interaction and data exchange with other satellite modules.

“Fig. 1” shows a block diagram of the hardware architecture.

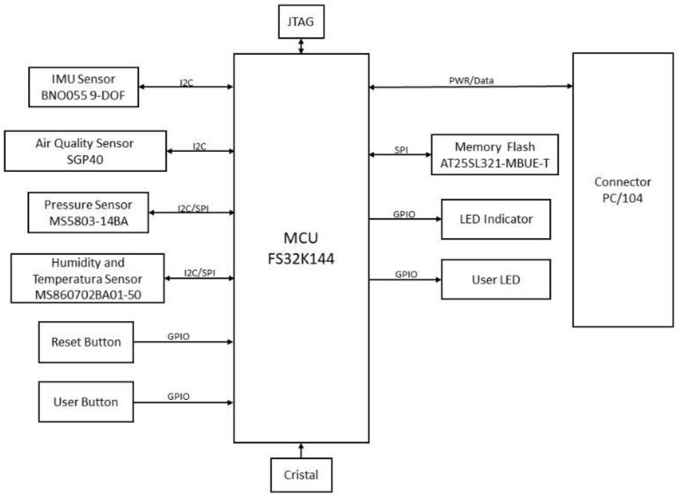

Fig. 1. EMIDDS-4 Hardware Architecture.

B) SOFTWARE ARCHITECTURE

The foundation of the EMIDSS-4 software architecture is based on AUTOSAR 4.4, a standard used in the automotive industry to distinguish the highest abstraction level between three software layers: application software (ASW), runtime environment (RTE), and basic software (BSW) which run on a microcontroller [4]. The main scope of the software in this project is to control the information collected and stored in memory by the wake-up and sleep mode sensors. The ASW layer contains the algorithm that manages the sensor readings and stores the data in a flash memory. The architecture does not use the RTE layer, instead it uses the Interface (IF) module to connect SWC with the BSW, and then BSW uses the module SERV to communicate with the Microcontroller Abstraction Layer (MAL) module to modify registers as shown in “Fig. 2”.

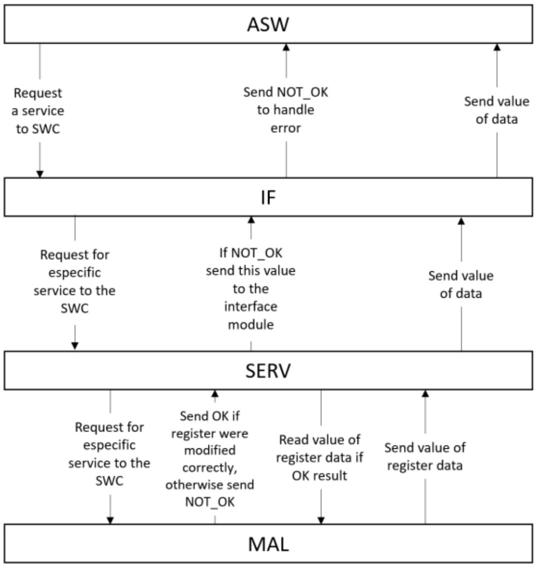

Fig. 2. Software Architecture EMIDDS-4

A. Application Software (ASW)

This module controls the read sequences of the information from the sensors. The application layer of the project includes a comprehensive suite of sensors and a memory described in Table I, such as an Inertial Measurement Unit (IMU), air quality, temperature, humidity and pressure sensors, and a flash memory; all consolidated within a software component module. This module serves as a central hub for data acquisition, processing, and saving data by enabling the precise tracking and monitoring device motion and various environmental factors. Seamless communication with the IF block of the basic software module guarantees efficient data transfer and streamlined integration, allowing the system to generate real-time insights.

TABLE I. PAYLOAD SENSORS

ITEM	PART NUMBER	COMMUNICATION PROTOCOL
IMU	BNO55 9-DOF	I2C
TEMP AND HUM. SENSOR	MS860702BA01-50	I2C
AQS	SGP40	I2C
PRESSURE SENSOR	MS5803-14BA	I2C
FLASH MEMORY	SST26VF032B	SPI

The block diagram shown in “Fig. 3” illustrates the flow of the project's application layer design. Initially, the microcontroller peripherals, memory, and four sensors specific to this project are initialized. Subsequently, at regular one-minute intervals, the microcontroller's integrated Real-Time Clock (RTC) generates an interruption, causing the microcontroller to awaken and read each sensor. The collected data is internally stored within a data structure, and once all the necessary data is gathered, it is transferred and stored on a 32 Mbit flash memory. To prevent data loss, the corresponding algorithm regularly checks if the memory is full. If it reaches full capacity, the algorithm begins overwriting the data from the starting point. However, based on the expected data volume and the memory's capacity, it is projected that there will be no need for overwriting the memory. Following the successful storage of information, the microcontroller reverts to a sleep mode until the next RTC interrupt is received, allowing the algorithmic flow to continue working seamlessly.

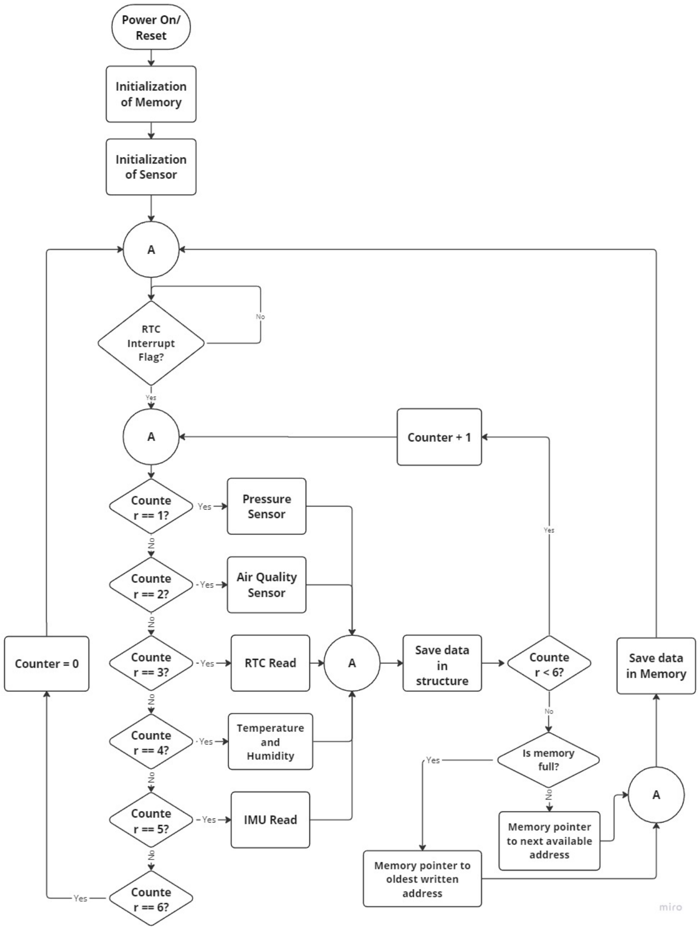

Fig. 3. Software Implementation Flow Diagram.

B. Basic Software (BSW)

The BSW contains the module interfaces that are independent of the microcontroller and the driver for every sensor by connecting with the MAL to send the command and establish communication. Every sensor has its own module to control the initialization and deinitialization states. These modules pass by reference a structure with temperature, pressure, and humidity values to be processed by ASW. Then, BSW is divided into different modules, each one handling a different nature of tasks:

- COM module - handles the interface between I2C, UART and SPI protocols. Within the COM module, the baud rate, stop bit, chip select, or DMA control can be changed.
- MEM module - contains the flash memory driver. After sensors send the value measurement, the MEM module stores the information in blocks of 32 bits.
- IO module - handles all the digital input and output peripherals.

C. *Microcontroller Abstraction Layer (MAL)*

The microcontroller abstraction layer contains the driver abstraction for every register of the microcontroller. It is highly dependent on the microcontroller provided by the NXP library. In MAL, all General Purpose I/O registers (GPIO) are set depending on their alternative function that enables the mode for the corresponding input/output as described in Table II.

TABLE II. GPIO CONFIGURATIONS

MC PIN	PORT/PIN	PERIPHERAL	SIGNAL
22	D7	LPUART2	TX
23	D6	LPUART2	RX
48	A2	LPI2C0	SDA
47	A3	LPI2C0	SCL
18	B5	LPSPI0	CE
19	B4	LPSPI0	MOSI
31	B3	LPSPI0	MISO
32	B2	LPSPI0	SCK
25	C1	GPIO	USER LED
43	B12	GPIO	AIR QUALITY SENSOR ENABLE
44	D4	GPIO	IMU SENSOR ENABLE
45	D3	GPIO	PRESSURE SENSOR ENABLE
46	D2	GPIO	HUMIDITY SENSOR ENABLE
1	D1	GPIO	PUSH BUTTON USER

D. *Hardware Test*

Once the ASW was defined, a separate code was developed to monitor the interaction between the hardware and software, where the hardware test originated. This program includes routines to test the peripherals inputs, outputs, and communication buses. It changes as the integrated button on the card is pressed, transitioning between operational states, and a sequence of LED blinking indicates the current state of the board.

III. RESULTS

The PCB shown in “Fig 4” was subsequently transported to the IPN laboratory for integration into the EMIDSS-4 “Fig 5”.

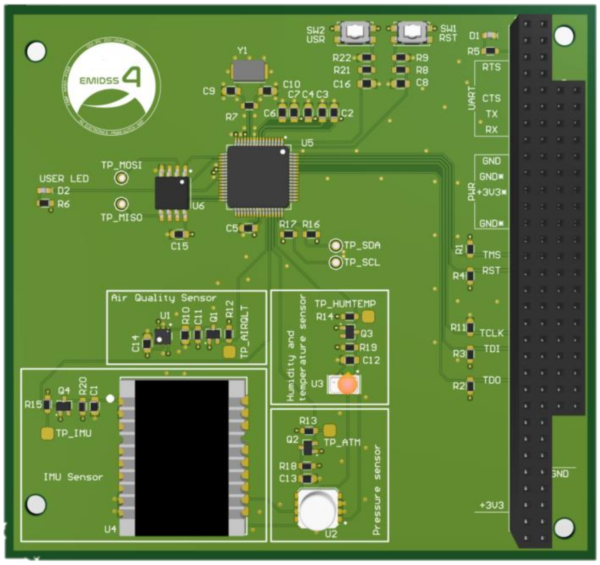

Fig. 4. Printed Circuit Board.

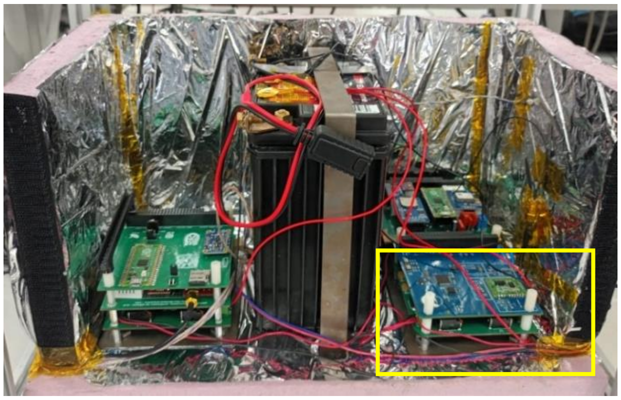

Fig. 5. EMIDSS-4 with the integrated PCB.

Communication between the microcontroller and sensor by the I2C protocol was successful, as evidenced by the acknowledge bit sent to the microcontroller shown in “Fig. 6”. This demonstrates that communication is established between the devices.

Type	Start	Duration
start	1.810 844 568 s	8 ns
address	1.810 854 560 s	89.48 µs
data	1.810 949 056 s	89.472 µs
stop	1.811 049 056 s	8 ns
start	1.811 055 024 s	8 ns
address	1.811 065 024 s	89.48 µs
data	1.811 159 520 s	89.472 µs
stop	1.811 260 272 s	8 ns
start	2.310 679 560 s	8 ns
address	2.310 689 552 s	89.48 µs
data	2.310 784 040 s	78.976 µs
data	2.310 868 024 s	89.48 µs

Fig. 6. I2C Frame Communication.

IV. CONCLUDING REMARKS

In conclusion, the study presents a proposal in the domain of data acquisition by developing a dependable and scalable hardware and software layered architecture. The transition from FPGA to a microcontroller-based solution has successfully addressed the portability, reliability, scalability of the previous architecture, resulting in continuous and adaptable data acquisition capabilities. The seamless integration of diverse sensors and the implementation of a layered software architecture have facilitated a highly efficient modular approach in managing the control, storage, and processing of sensor data. This research contributes to the development of nanosatellites by establishing a foundation for future studies and implementation in data acquisition methods, creating new opportunities for scientific research.

REFERENCES

[1] W. Shiroma *et al.*, "CubeSats: A bright future for nanosatellites," *Open Engineering*, vol. 1, no. 1, 2011. doi:10.2478/s13531-011-0007-8 "Small satellites," SpaceTech, https://www.spacetech.global/small-satellites (accessed Jun. 12, 2023).

[2] "Small satellites," SpaceTech, https://www.spacetech.global/small-satellites (accessed Jun. 16, 2023).

[3] B. P. Acosta, E. Angel Guzmán, J. A. Osuna Valdez, L. Rizo Domínguez, M. Mendoza Barcenas and M. I. R. Prieto Meléndez, "Adaptable O-Board Computer for Nanosatellites," *2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, NY, USA, 2022, pp. 0421-0425, doi: 10.1109/UEMCON54665.2022.9965674.

[4] "S32K1 general-purpose mcus," S32K1 General-Purpose MCUs | NXP Semiconductors, https://www.nxp.com/products/processors-and-microcontrollers/s32-automotive-platform/s32k-auto-general-purpose-mcus/s32k1-microcontrollers-for-automotive-general-purpose:S32K1 (accessed Jan. 16, 2023).

[5] "SGP40 data sheet - sensirion," Indoor Air Quality Sensor for VOC Measurements, https://sensirion.com/media/documents/296373BB/6203C5DF/Sensirion_Gas_Sensors_Datasheet_SGP40.pdf (accessed Jun. 16, 2023).

[6] "MS860702BA01-50 : Meas Board Mount Pressure Sensors," TE Connectivity, https://www.te.com/usa-en/product-MS860702BA01-50.html (accessed Jun. 16, 2023).

[7] "MS5803-14BA miniature 14 bar module," TE connectivity, https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5803-14BA%7FB3%7Fpdf%7FEnglish%7FENG_DS_MS5803-14BA_B3.pdf%7FCAT-BLPS0013 (accessed Jun. 16, 2023).

[8] "FSM30X," CEVA, https://www.ceva-dsp.com/wp-content/uploads/2019/10/FSM30x-Datasheet.pdf (accessed Jun. 16, 2023).