People's Democratic Republic of Algeria

وزارة التـــعلـــيـــــم العـــــالـــــي والبـحـــــــث العـلمـــــي
Ministry of Higher Education and Scientific Research

جامعة وهران 2 محمد بن أ حمد
University of Oran 2 Mohamed Ben Ahmed

معهد الصيانة والأمن الصناعي
Institute of Industrial Maintenance and Safety

Department of Instrumentation Maintenance

Master's thesis

For the attainment of the Master's degree.

Field of study: Industrial Engineering

Specialization: Automation and Instrumentation Maintenance

Topic

Realization and digital twin development of a Smart Fuel cell system.

Presented and publicly defended by:

Bellal Yasmine

Before the jury composed of:

First and last name	Grade	Establishment	Quality
Tarfaya Anis	MCB	IMSI-Univ. Of Oran2	**President**
Metahri Dhiyaeddine	MCA	IMSI-Univ. Of Oran2	**Supervisor**
Bouchaala Mohammed Abdelghani	MCB	IMSI-Univ. Of Oran2	**Examiner**

***Academic year**: 2025 / 2026*

وقل رب زدني علما



≪ Realization and Digital Twin Development of a Smart Fuel Cell System≫

Abstract

Fuel cell performance assessment and methane monitoring are critical for promoting sustainable energy solutions. The goal of our project is to create a Solid Oxide Fuel Cell (SOFC) Digital Twin Prototype (DTP) that is connected to an Internet of Things (IoT)-based methane monitoring system. We collect methane concentration data using an ESP32 microcontroller and a MQ-4 gas sensor, then utilize the MQTT communication protocol to relay the data in real time. These data are processed using a dynamic SOFC model created in MATLAB/Simulink to estimate fuel cell performance, including voltage, losses, temperature, and electrical power generation. To enable real-time visibility and interaction between the virtual and physical environments, the estimated power is also transmitted back to the monitoring system. The obtained results confirm the feasibility of combining IoT technologies with Digital Twin concepts for real-time monitoring and performance estimation, providing a solid foundation for future developments involving real SOFC systems and advanced monitoring solutions.

Keyword: Fuel cell, Sustainable energy, Solid Oxide Fuel Cell, Digital Twin Prototype, Internet of Things.

ملخص

يعد تقييم أداء خلايا الوقود ومراقبة غاز الميثان أمرا بالغ الأهمية لدعم حلول الطاقة المستدامة. يهدف مشروعنا إلى إنشاء نموذج أولي رقمي لخلية الأكسيد الصلب (SOFC) متصل بنظام مراقبة غاز الميثان القائم على انترنيت الأشياء (IoT). نجمع بيانات تركيز الميثان باستخدام وحدة تحكم دقيقة ESP32 ومستشعر الغاز MQ-4، ثم نستخدم بروتوكول اتصال MQTT لنقل البيانات في الوقت الفعلي. تعالج هذه البيانات باستخدام نموذج ديناميكي لخلية وقود أكسيد الصلب (SOFC) تم إنشاؤه في MATLAB/Simulink لتقدير أداء الخلية، بما في ذلك الجهد، الفقد، درجة الحرارة، والطاقة الكهربائية. ولتمكين الرؤية والتفاعل في الوقت الفعلي بين البيئتين الافتراضية والمادية، ترسل الطاقة المقدرة أيضا إلى نظام المراقبة. تؤكد النتائج المتحصل عليها جدوى دمج تقنيات إنترنيت الأشياء مع مفاهيم التوأم الرقمي للمراقبة في الوقت الفعلي وتقدير الأداء، مما يوفر أساسا متينا للتطورات المستقبلية التي تشمل أنظمة خلايا وقود أكسيد الصلب (SOFC) حقيقية وحلول مراقبة متقدمة.

كلمة مفتاحية: خلية وقود، الطاقة المستدامة، خلية وقود أكسيد الصلب، أنترنيت الأشياء.

Résumé

L'évaluation des performances des piles à combustible et la surveillance du méthane sont essentielles pour renforcer la sécurité industrielle et promouvoir des solutions énergétiques durables. Notre projet vise à créer un prototype de jumeau numérique (DTP) de pile à combustible à oxyde solide (SOFC) connecté à un système de surveillance du méthane basé sur l'Internet des objets (IoT). Nous collectons les données de concentration de méthane à l'aide d'un microcontrôleur ESP32 et d'un capteur de gaz MQ-4, puis utilisons le protocole de communication MQTT pour transmettre ces données en temps réel. Ces données sont traitées par un modèle dynamique de SOFC créé sous MATLAB/Simulink afin d'estimer les performances de la pile, notamment la tension, les pertes, la température et la production d'énergie électrique. Pour permettre une visibilité et une interaction en temps réel entre les environnements virtuel et physique, la puissance estimée est également renvoyée au système de surveillance. Les résultats obtenus confirment la faisabilité de la combinaison des technologies IoT et des concepts de jumeau numérique pour la surveillance en temps réel et l'estimation des performances, jetant ainsi les bases de futurs développements impliquant des systèmes SOFC réels et des solutions de surveillance avancées.

Mot-clé : Pile à combustible, Energie Durable, Pile à combustible à oxyde solide, jumeau numérique, Internet des objets.

Dedication

*I dedicate this work to my small **family**,*

*my dear **mother** who supported me every step of the way, and to my little **sister**.*

*And I mustn't forget the first **engineer** I discussed the idea with, who encouraged me to work on it may God have mercy on him. Today, I say that I have completed the first step, which was this research.*

Acknowledgement

First and foremost, I would like to thank ***ALLAH*** *for granting me success throughout my Academic journey and for providing me with the strength, determination, and Perseverance needed to complete this research.*

I would also like to express my sincere gratitude to all my ***professors****.*

I thank especially my supervisor, ***Mr. METAHRI DHIYAEDDINE****, for his Valuable guidance, and encouragement throughout this work. His belief in This idea motivated me to develop it further and explore its potential through Modern technologies.*

Finally, I would like to express my sincere gratitude to ***everyone*** *who contributed Directly or indirectly, to the completion of this thesis through their support, Assistance, and valuable advice.*

Yasmine.

Table of Contents

Abstract ..... **I**

I ..... **ملخص**

Résumé ..... **I**

Dedication ..... **II**

Acknowledgement ..... **III**

Lists of figures ..... **IX**

Lists of tables ..... **XI**

Abbreviation ..... **XII**

General Introduction ..... **XIV**

CHAPTER I: Generalities On Fuel Cells

I.1 Introduction **1**

I.2 History of fuel cell **1**

I.3 Operating principles of fuel cell **2**
I.3.1 General architecture of a single cell 3
I.3.1.1 The membrane (or electrolyte) 3
I.3.1.2 The electrodes 3
I.3.1.3 The bipolar & collector plates 4
I.3.2 Electrochemical energy conversion of a fuel cell 5
I.3.2.1 Hydrogen Oxidation Reaction (HOR) 6
I.3.2.2 Oxygen Reduction Reaction (ORR) 6
I.3.2.3 Overall reaction occurring in a fuel cell 7

I.4 Classification of fuel cells **7**
I.4.1 Solid oxide fuel cell (SOFC) 8
I.4.1.1 SOFC charateristics 8
I.4.2 Molten carbonate fuel cell (MCFC) 9
I.4.2.1 MCFC characteristics 9
I.4.3 Phosphoric acid fuel cell (PAFC) 10
I.4.3.1 PAFC characteristics 11
I.4.4 Proton exchange membrane fuel cell (PEMFC) 11
I.4.4.1 PEMFC characteristics 12
I.4.5 Alkaline fuel cell (AFC) 12
I.4.5.1 AFC characteristics 12

I.5 Advantages and disadvantages of fuel cells 13

I.6 Technological choice of SOFC 15
I.6.1 Conventional fuel cells material 15
I.6.1.1 Electrolyte 15
I.6.1.2 Anode 16
I.6.1.3 Cathode 16
I.6.2 SOFC structure 16
I.6.2.1 Tubular structure 16
I.6.2.2 Planar structure 17
I.6.3 Operation of a SOFC under methane 17

I.7 Electrochemical modeling of a solid oxide fuel cell (SOFC) 19
I.7.1 Theoretical potential of a fuel cell (Nernst equation) 19
I.7.2 Actual cell voltage 20
I.7.2.1 Activation losses 21
I.7.2.2 Ohmic losses 22
I.7.2.3 Concentration losses 22

I.8 Existing research on fuel cells 23
I.8.1 PEM fuel cell modeling for transportation applications 23
I.8.2 Performance Evaluation of Hydrogen Fuel Cell Vehicles 23
I.8.3 Dynamic SOFC modeling for power system simulation 23
I.8.4 MATLAB/Simulink implementation of a dynamic SOFC model 24
I.8.5 Experimental study of internal methane reforming in Biogas fed SOFCs 24
I.8.6 Methane rich internal reforming in SOFC stacks: Performance evaluation 24

I.9 Conclusion 25

CHAPTER II: Generalities On Digital Twin

II.1 Introduction 24

II.2 History and origin of digital twin 24

II.3 Conceptual and technical foundations of the digital twin 25
II.3.1 Concept and principals of the digital twin 25
II.3.2 Types of digital twins 26
II.3.2.1 Digital twin instance (DTI) 26
II.3.2.2 Digital twin prototype (DTP) 26
II.3.2.3 Performance digital twin (PDT) 27
II.3.3 Integration levels of digital twin 27
II.3.3.1 Digital model 28
II.3.3.2 Digital shadow 28
II.3.3.3 Digital Twin 28
II.3.4 Architecture of digital twin 29
II.3.4.1 Physical component 29
II.3.4.2 Virtual component 30
II.3.4.3 Communication system (IoT & Sensors) 30
II.3.4.4 Real-Time synchronization and data flow 30

II.4 Industry 4.0 and digital transformation **31**

II.5 Benefits and applications of digital twin **31**

II.6 Associated technologies in digital twin **32**
II.6.1 Internet of Things (IoT) 32
II.6.2 Simulation engines 32
II.6.2.1 ANSYS Twin Builder 32
II.6.2.2 Simcenter (Siemens) 33
II.6.2.3 MATLAB/Simulink 33
II.6.2.3.1 Choice of MATLAB/Simulink for modeling 33
II.6.3 MQTT communication protocol 33
II.6.3.1 MQTT architecture (broker, publisher, and subscriber) 34
II.6.3.2 MQTT Qos Levels 34
II.6.3.3 MQTT vs Other IoT Protocols 35

II.7 Existing research on digital twin of fuel cell **36**
II.7.1 Digital twin for Proton Exchange Membrane Fuel Cells (PEMFC) 36
II.7.2 Digital twin for Solid Oxide Fuel Cells (SOFC) 37
II.7.3 IoT-Based digital twin for energy system 37

II.8 Research gap and contribution of our work **37**

II.9 Conclusion **39**

CHAPTER III : Smart Fuel Cell Monitor Realization

III.1 Introduction **42**

III.2 Overall System Overview: **42**

III.3 Function of Our Smart Fuel Cell Monitoring System **43**

III.4 Hardware and software components used in our monitoring system **44**
III.4.1 Gas Sensor 44
III.4.1.1 MQ-4 Gas sensor 44
III.4.1.2 MQ-4 Operating with Methane 45
III.4.1.3 The Pins Used in Our MQ-4 Gas Sensor 45
III.4.1.4 Characteristics of the MQ-4 gas sensor 46
III.4.1.5 Wiring of the MQ-4 with The Microcontroller 46
III.4.2 ESP32 Microcontroller 47
III.4.2.1 Origin of the ESP32 Microcontroller: 47
III.4.2.2 The ESP32 Microcontroller family 47
III.4.2.3 The differences between ESP32 chips, modules, and development boards 48
III.4.2.3.1 ESP32 chips 48
III.4.2.3.2 ESP32 modules 48
III.4.2.3.3 ESP32 development boards 49
III.4.2.3.4 The comparison between ESP32 chips, modules, and development boards 49
III.4.2.3.5 Type of ESP32 used in our Smart Fuel Cell Monitoring System 50
III.4.2.3.6 Function of The NodeMCU ESP32-S in the Monitoring System 51
III.4.2.3.7 Technical Characteristics of the NodeMCU-32S 51

III.4.2.3.7.1 Specifications 51
III.4.2.3.7.2 NodeMCU-32S Pins definitions 52
III.4.3 Buzzer and LED 52
III.4.3.1 Buzzer 53
III.4.3.1.1 Characteristics of the active buzzer 54
III.4.3.1.2 Wiring of the buzzer with ESP32S 54
III.4.3.2 Light Emitting diodes (LED) 55
III.4.3.2.1 Characteristics of the LED 55
III.4.3.2.2 Wiring of the LED with the ESP32S 56
III.4.4 Arduino IDE 56
III.4.4.1 Adding ESP32 support to Arduino IDE 57
III.4.4.2 Why we chose Arduino IDE 57
III.4.5 MQTT Protocol used in our Smart Fuel Cell Monitoring System 58
III.4.5.1 Mosquitto Broker Used 58
III.4.6 Complete wiring table of the Smart Fuel cell Monitor 59

III.5 Conclusion 62

CHAPTER IV : Digital Twin Of SOFC

IV.1 Introduction 64

IV.2 Digital Twin Prototype of SOFC in MATLAB/Simulink 64

IV.3 Enhanced Dynamic Modeling Approach for SOFC Systems 66
IV.3.1 Elements Adopted from the Reference SOFC Dynamic Model 66

IV.4 Electrochemical Modeling of SOFC in MATLAB/Simulink 68
IV.4.1 Stack design parameters 68
IV.4.2 Functional Blocks of the SOFC Digital Twin Prototype 68
IV.4.3 Organization of the Functional Blocks of the SOFC Digital Twin Prototype 69
IV.4.3.1 Input layer 69
IV.4.3.1.1 Level-2 MATLAB S-Function Block 69
IV.4.3.1.2 Pre-reformer Block 70
IV.4.3.1.3 Block of Calc-FuelFr 72
IV.4.3.1.4 Block of calc-AirFr 73
IV.4.3.1.5 Partial Pressures block 74
IV.4.3.1.6 SOFC Voltage cell Block (SOFC Vcell) 77
IV.4.3.1.7 Thermal SOFC block 79
IV.4.3.1.8 Total stack power block 82
IV.4.3.1.9 Publish power block 82
IV.4.3.1.10 Scopes blocks 82

IV.5 General System Logic 83
IV.5.1 MATLAB Communication Scripts 83
IV.5.1.1 Shared Data Management: 83
IV.5.1.2 Simulink Interface Level-2 MATLAB S-Function 84
IV.5.1.3 IoT Data Acquisition and Communication Layer 84
IV.5.2 Fuel Cell Digital Twin Operation 84
IV.5.2.1 Methane Detection and Data acquisition 85

IV.5.2.2 Data Transmission Via MQTT Broker 85
IV.5.2.3 Data Transmission Via MQTT Broker 86
IV.5.2.4 Digital Twin Simulation 86
IV.5.2.5 Feedback from the SOFC Digital Twin 87
IV.5.2.6 Results and Discussion 88
IV.5.2.6.1 Nernst Voltage and Fuel Cell Voltage 88
IV.5.2.6.2 Activation, Concentration, and Ohmic losses 89
IV.5.2.6.3 Temperature 90
IV.5.2.6.4 Electrical Power 91

IV.6 Future Work **91**

IV.7 Conclusion **92**

GENERAL CONCLUSION **93**

REFERENCES **94**

LISTS OF FIGURES

Chapter I:

Figure I.2 Sir W. Grove [12]..... 1
Figure I.1 Fuel cell of Grove [12]. 1
Figure I.3 William Robert Grove presenting one of his first fuel cells [12]..... 2
Figure I.4 Stack d'une pile de type PEM [Clefs CEA] [13]..... 2
Figure I.5 Exploded view of PEM fuel cell stack [79]..... 3
Figure I.6 Main cell components (not in scale) and processes [14]. 4
Figure I.7 First step: Hydrogen decomposition at the anode (adopted from [12])..... 6
Figure I.8 Second step: Oxygen decomposition at the cathode (adopted from [12]). 6
Figure I.9 Overall operating diagram (adopted from [12]). 7
Figure I.10 SOFC schematic diagram [72]..... 8
Figure I.11 Schematic diagram of a MCFC [72]..... 9
Figure I.12 Schematic diagram of PAFC [72]..... 10
Figure I.13 Schematic diagram of PEMFC [72].....11
Figure I.14 Schematic diagram of AFC [72]. 12
Figure I.15 Tubular SOFC design [73]..... 17
Figure I.16 Palanar SOFC design [17]. 17
Figure I. 17 Polarization curve (V–I) of a solid oxide fuel cell (SOFC) [20]. 21

Chapter II :

Figure II.1 Enabling technologies for cognizing and controlling physical word [39]..... 25
Figure II.2 Data flow in digital model [35]. 28
Figure II.3 Data flow in a digital shadow [35]. 28
Figure II.4 Data flow in a digital twin [35]. 29
Figure II.5 Digital Twin architecture [34]. 29
Figure II.6 MQTT Publish/Subscribe Architecture [49]..... 34
Figure II.7 Protocol Stack for IoT Systems [48]. 35

Chapter III :

Figure III. 1 Overall implementation of the Proposed Smart Fuel Cell Monitor and SOFC DT..... 42
Figure III. 2 Block Diagram of The Smart fuel Cell Monitoring System..... 43
Figure III. 4 MQ-4 Gas Sensor. 45
Figure III. 5 Internal construction of a Metal Oxide Semiconductor MQ-series gas sensor [78]. 45
Figure III. 6 Interfacing of MQ-4 Gas Sensor with ESP32. 46
Figure III. 7 Wiring diagram for interfacing MQ-4 with ESP32S using WOKWI Simulator. 47
Figure III. 8 Hierarchy of ESP32 Hardware integration [60]. 49
Figure III. 9 NodeMCU ESP32-S development board. 51
Figure III. 10 NodeMCU-32S Pins [61]. 52
Figure III. 11 A typical actuating unit [62]. 53
Figure III. 13 Difference between Active and Passive buzzer. 54
Figure III. 12 Active buzzer. 54
Figure III. 11 Wiring diagram for interfacing Active buzzer with NodeMCU ESP32-S 55
Figure III. 12 Internal structure and components of a light emitting diode. 55
Figure III. 16 Wiring Diagram for Interfacing LED with NodeMCU ESP-32S..... 56

Figure III. 14 Arduino IDE software interface. 57
Figure III. 16 Communication Architecture of Our Smart Fuel Cell Monitoring System. 59
Figure III. 17 Wiring Diagram for Our Smart Fuel Cell using The WOKWI Simulator. 60
Figure III. 18 The Fuel Cell Monitor Prototype. 60
Figure III. 19 Smart Fuel Cell Monitor Code Flowchart. 62

Chapter IV :

Figure IV. 1 SOFC Model in MATLAB/Simulink. 65
Figure IV. 2 Pre-reformer Block Flowchart Code. 71
Figure IV. 2 Pre-reformer Block Flowchart Code. 71
Figure IV. 3 Calc-FuelFr Block Flowchart Code. 73
Figure IV. 4 Calc-AirFr Block Flowchart Code. 74
Figure IV. 5 Partial Pressures Block Flowchart Code. 76
Figure IV. 5 Partial Pressures Block Flowchart Code. 76
Figure IV. 6 SOFC Vcell Block Flowchart Code. 78
Figure IV. 7 Thermal SOFC Block Flowchart Code. 81
Figure IV. 8 Publish Power Block Flowchart Code. 82
Figure IV. 9 Simulation of Methane using a Lighter. 85
Figure IV. 10 The Mosquitto Broker Interface and the Transmitted Data. 86
Figure IV. 11 The transmitted Data in the Command Window of MATLAB. 86
Figure IV. 12 SOFC DT Model Currently Running. 87
Figure IV. 13 LED at High Brightness Due to Methane Detection. 87
Figure IV. 14 Real-Time Simulation Results of the SOFC Digital Twin. 88
Figure IV. 14 Real-Time Simulation Results of the SOFC Digital Twin. 88
Figure IV. 15 The Behavior of the Enernst and Vfc curves. 89
Figure IV. 16 The Behavior of the Activation, Concentration, and Ohmic Losses. 90
Figure IV. 17 The Temperature behavior curve of SOFC Model. 90
Figure IV. 18 The Electrical Power Behavior Curve of the SOFC Model. 91

LISTS OF TABLES

Chapter I :

Table I.1 Eight simultaneous processes in a fuel cell (adapted from Barbir [14]). 5
Table I.2 SOFC characteristics [12, 14, 15]. 9
Table I.3 MCFC characteristics [12, 14, 15]. 10
Table I.4 PAFC characteristics [12, 14, 15]. 11
Table I.5 PEMFC characteristics [12, 14]. 12
Table I.6 AFC characteristics [12, 14, 15]. 13
Table I.7 Table of advantages and disadvantages of different types fuel cells [12]. 13
Table I.8 Parameters and their significations [2, 19]. 20

Chapter II :

Table II.1 levels Classification of digital representations [35, 34]. 27
Table II. 2 MQTT Quality of Service levels [50, 51] 35
Table II.3 Comparison of MQTT with other IoT protocols [48, 51]. 36

Chapter III :

Table III. 1 Technical characteristics of Our MQ-4 sensor [56, 57]. 46
Table III. 2 Wiring connections and pin functions for the MQ-4. 46
Table III. 3 Main variants of the ESP32 family [59]. 48
Table III. 4 Comparaison of ESP32 Hardware formats [60]. 49
Table III. 5 Technical characteristics of the active buzzer [63]. 54
Table III. 6 Wiring connections and pin functions for the Active buzzer. 54
Table III. 7 Technical characteristics of standard 5 mm LED [65]. 56
Table III. 8 Wiring connections and pin functions for the LED. 56
Table III. 9 Wiring connections and pin functions for our Smart Fuel Cell realization. 59

Chapter IV :

Table IV. 1 Parameters of the SOFC system used in the model [2, 18, 66]. 67
Table IV. 2 Parameters that can be changed in our model. 67
Table IV. 3 The block names, types, and layers used in our SOFC model. 69
Table IV.4 The Parameters used in the Pre-reformer Block. 72
Table IV.5 Parameters and their significations [66]. 79

ABBREVIATION

AFC: Alkaline Fuel Cell

AMQP: Advanced Message Queuing Protocol

ARO: Artificial Rabbits Optimization

BLE: Bluetooth Low Energy

CAD: Computer Aided Design

CFD: Computational Fluid Dynamics

CNN: Convolutional Neural Network

CoAP: Constrained Application Protocol

CPU: Central Processing Unit

DC: Direct Current

DIR: Direct Internal Reforming

DT: Digital Twin

DTI: Digital Twin Instance

DTP: Digital Twin Prototype

FEM: Finite Element Models

GE: General Electric

GPIO: General Purpose Input/Output

HOR: Hydrogen Oxidation Reaction

HTTP: Hypertext Transport Protocol

I2C: Inter Integrated Circuit

IoT: Internet of Things

IUPAC: International Union of Pure and Applied Chemistry

LED: Light Emitting Diode

LSM: Lanthanum Strontium Manganite

LSTM: Long Short-Term Memory

MCFC: Molten Carbonate Fuel Cell

MEA: Membrane Electrode Assembly

ML: Machine Learning

MQTT: Message Queuing Telemetry Transport

NASA: National Aeronautics and Space Administration

Ni-GDC: Nickel-Gadolinium Doped Ceria

Ni-YSZ : Nickel-Yttria Stabilized Zirconia

NN: Neural Network

ORR: Oxygen Reduction Reaction

PAFC: Phosphoric Acid Fuel Cell

PDT: Performance Digital Twin

PEM: Proton Exchange Membrane

QoS: Quality of Service

RISC-V: Reduced Instruction Set Computing-Five

RUL: Remaining Useful Life

SDIO: Secure Digital Output

SoC: System on Chip

SOFC: Solid Oxide Fuel Cell

SR: Steam Reforming

SSL: Secure Sockets Layer

TCP: Transmission Control Protocol

TL: Transfer Learning

TLS: Transport Layer Security

U.S: United State

UART: Universal Asynchronous Receive/Transmitter

UDP: User Datagram Protocol

WGR: Water Gas Reaction

XGBoost: eXtreme Gradient Boosting

General Introduction

The reduction of greenhouse gas emissions has become one of the greatest global environmental challenges. Among these gases, methane (CH_4) is of particular concern because its global warming potential is significantly higher than that of carbon dioxide [1]. Large amounts of methane are released from industrial activities, particularly in the oil and gas sector, where part of this valuable resource is still vented or flared. Converting methane into electricity not only helps reduce environmental impact but also enables the recovery of an important energy source.

Fuel cells have emerged as a promising clean energy technology because they convert the chemical energy of a fuel directly into electricity with high efficiency and low emissions. Several types of fuel cells exist, including Molten Carbonate (MCFC), Polymer Electrolyte Membrane (PEMFC), Phosphoric Acid (PAFC), Solid Oxide (SOFC), and Alkaline Fuel Cells (AFC). Among them, Solid Oxide Fuel Cells are particularly attractive due to their high efficiency, fuel flexibility, and ability to operate with fuels derived from methane [2]. However, their performance strongly depends on operating conditions such as fuel composition, temperature, and reactant flow rates, making continuous monitoring and analysis essential.

At the same time, Digital Twin technology has become a key component of Industry 4.0, enabling real-time monitoring, simulation, and optimization of complex engineering systems through continuous data exchange between physical and virtual models [3, 4]. In fuel cell applications, Digital Twins provide valuable tools for understanding system behavior, improving performance, and supporting predictive maintenance.

Although many studies have investigated Digital Twins for fuel cells, most focus on Polymer Electrolyte Membrane Fuel Cells for degradation monitoring, remaining useful life prediction, and performance optimization [5, 6, 7]. In contrast, research on Solid Oxide Fuel Cell Digital Twins remains limited, with existing works mainly addressing parameter estimation or voltage prediction instead of developing real-time Digital Twin prototypes connected to live measurements [8, 9, 10].

To address this gap, our work presents the development of a real-time Digital Twin prototype for a Solid Oxide Fuel Cell to analyze the effect of methane concentration on fuel cell performance. The proposed framework combines an IoT-based Smart Fuel Cell Monitoring System with a physics-based SOFC model implemented in MATLAB/Simulink. Real-time methane measurements continuously update the Digital Twin, allowing dynamic analysis of SOFC behavior. Our work provides a practical proof of concept for integrating Digital Twin and IoT technologies into fuel cell systems and establishes a foundation for future implementation on real SOFC platforms.

Our thesis is organized into four chapters. Chapter I introduces the fundamentals of fuel cells and methane utilization. Chapter II presents Digital Twin technology and its applications to fuel cell systems. Chapter III describes the IoT communication architecture of the Smart Fuel Cell Monitoring System. Chapter IV presents the development of the SOFC Digital Twin prototype in MATLAB/Simulink and discusses the obtained results.

CHAPTER I:

Generalities on Fuel Cells

I.1 Introduction

Countries around the world, including Algeria, are now striving to establish precise systems for monitoring their carbon footprint and achieving a balance between energy production and environmental protection. This is being done by reducing methane emissions in gas facilities [11]. As methane accounts for 30% of global warming. Reducing this percentage is essential for achieving sustainability, mitigating global warming, and improving air quality [1].

In order to show that methane recovered from industrial facilities installations can be valued as electricity using the technologies of fuel cells. So, in this first chapter gives the general background needed to follow the rest of the thesis. We start by giving the historical development of fuel cell technology, its operating principle, as well as its types, advantages, and disadvantages. The type that we chose it is presented more specifically at the end of the chapter, together with a review of the existing research on fuel cells.

I.2 History of fuel cell

In 1839, Welsh-born physicist William Robert Grove conducted what historians believe to be the first real-world application of the fuel cell concept. Grove was originally trained as a lawyer, but he became interested in electrochemistry, specifically the inverse problem of electrolysis: could the recombination of hydrogen and oxygen be utilized to generate electricity instead of utilizing electricity to dissociate water? In response, he developed a gadget he dubbed a "gas battery" that was based on platinum electrodes submerged in diluted sulfuric acid [12]. The figures I.1, I.2, and I.3 illustrate the Sir William Grove and his first prototype of hydrogen fuel cell.

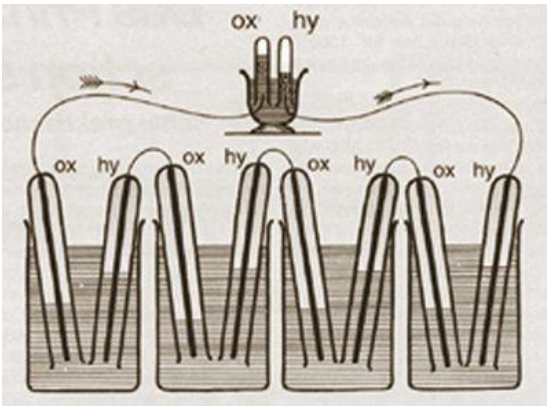


Figure I.1 Fuel cell of Grove [12].



Figure I.2 Sir W. Grove [12].

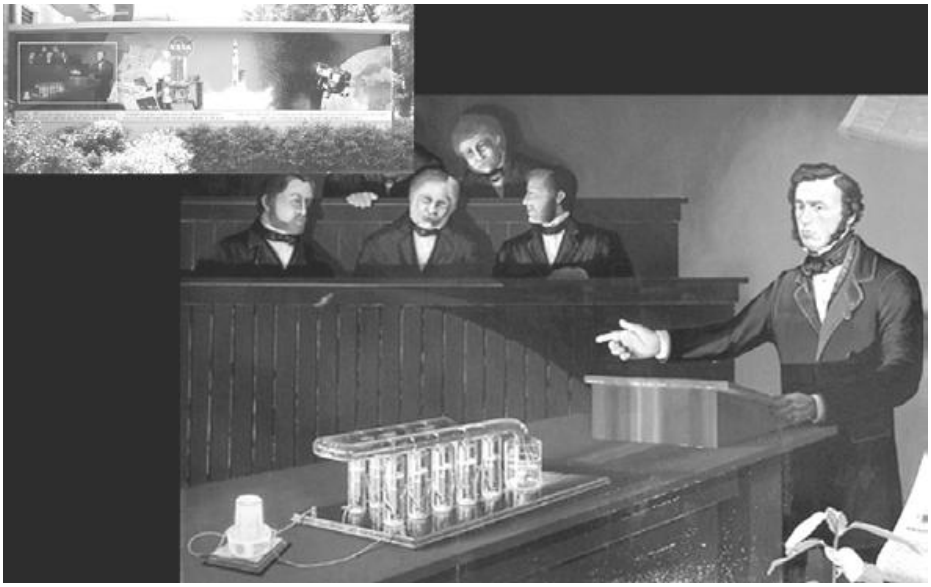


Figure I.3 William Robert Grove presenting one of his first fuel cells (source: Working Borough Council - Wall painting) [12].

At that time, General Electric and NASA worked on this technology for space missions. Grubb and Niedrach created a fuel cell that was used in Gemini 5, and later in Apollo and Space Shuttle missions. The fuel cell system was made of two parts, and each part could produce around 1 kW of electricity [12].

I.3 Operating principles of fuel cell

A single main cell in a fuel cell consists of an electrode-membrane-electrode (EME), which work together to convert the chemical energy into electrical energy and connecting these cells in series forms what is known as a stack [13].

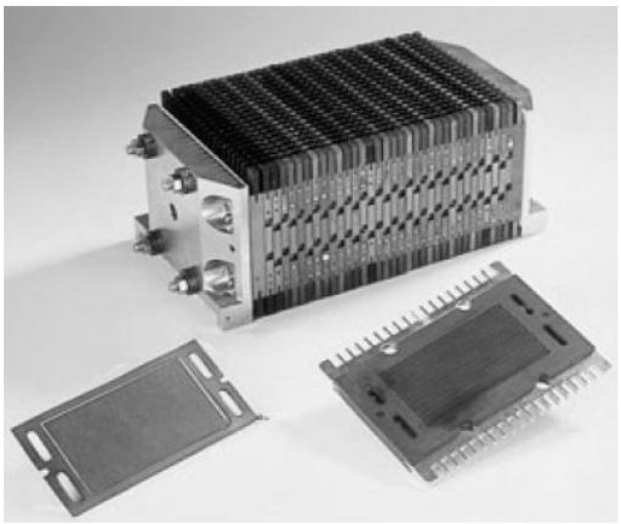


Figure I.4 Stack d'une pile de type PEM [Clefs CEA] [13].

I.3.1 General architecture of a single cell

The description of the cell architecture given here follows mainly Barbir [14], with complementary information from O'Hayre et al. [15]. From the inside out, a single cell is composed of the following layers: the electrolyte at the center then the two electrodes on each side, and the bipolar plates on the outside. Each of these layers has its own role and its own technical constraints, and the way they are stacked together is what determines the performance of the whole fuel cell. The figure I.5 illustrates an explored view of the PEM stack.

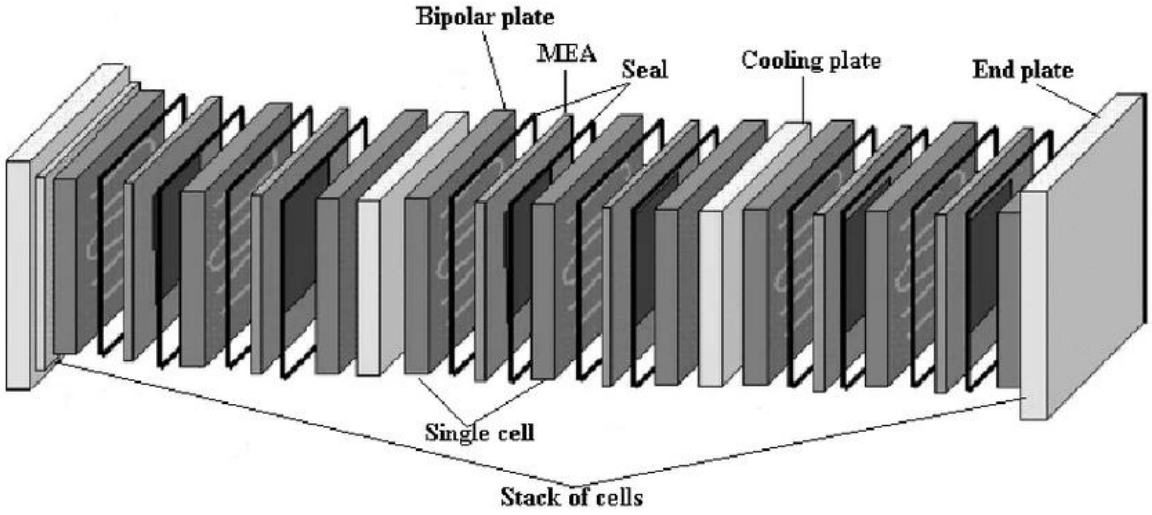


Figure I.5 Exploded view of PEM fuel cell stack [79].

I.3.1.1 The membrane (or electrolyte)

The electrolyte is the core of cell, a proton-conductive polymer membrane separates the anode side from the cathode side [14]. Its function is double: it has to let the ions cross from one electrode to the other, and at the same time it has to block the electrons and the gases. The electrons are forced to take an external path, which is exactly how the electrical current is collected. If the membrane were permeable to electrons or to the reactant gases, the cell would short-circuit chemically and the voltage would collapse.

I.3.1.2 The electrodes

On both sides of the membrane there is a porous electrode. The porosity is essential because the reactant gases are fed from the back of the electrode and have to reach the catalyst layer where the electrochemical reaction actually takes place [14, 15]. The catalyst layer is a very thin region, sometimes considered as part of the membrane and sometimes as part of the electrode depending on the manufacturing technique. The whole stack of membrane plus the

two electrodes is called the **Membrane Electrode Assembly**, abbreviated MEA, and it is the active core of the cell.

I.3.1.3 The bipolar & collector plates

Outside the MEA, the cell is closed by two plates that have several jobs at the same time [14]. They contain the channels through which the reactant gases are distributed over the surface of the electrode. They collect the electrical current produced by the cell. They also conduct heat to or from the cooling fluid. In a stack of several cells, the same plate plays the role of cathode collector for one cell and anode collector for the next one, which is why the term *bipolar plate* is used [15].

Inside the fuel cell, many different processes occur at the same time, and they are all interconnected, and this what the figure I.6 illustrates.

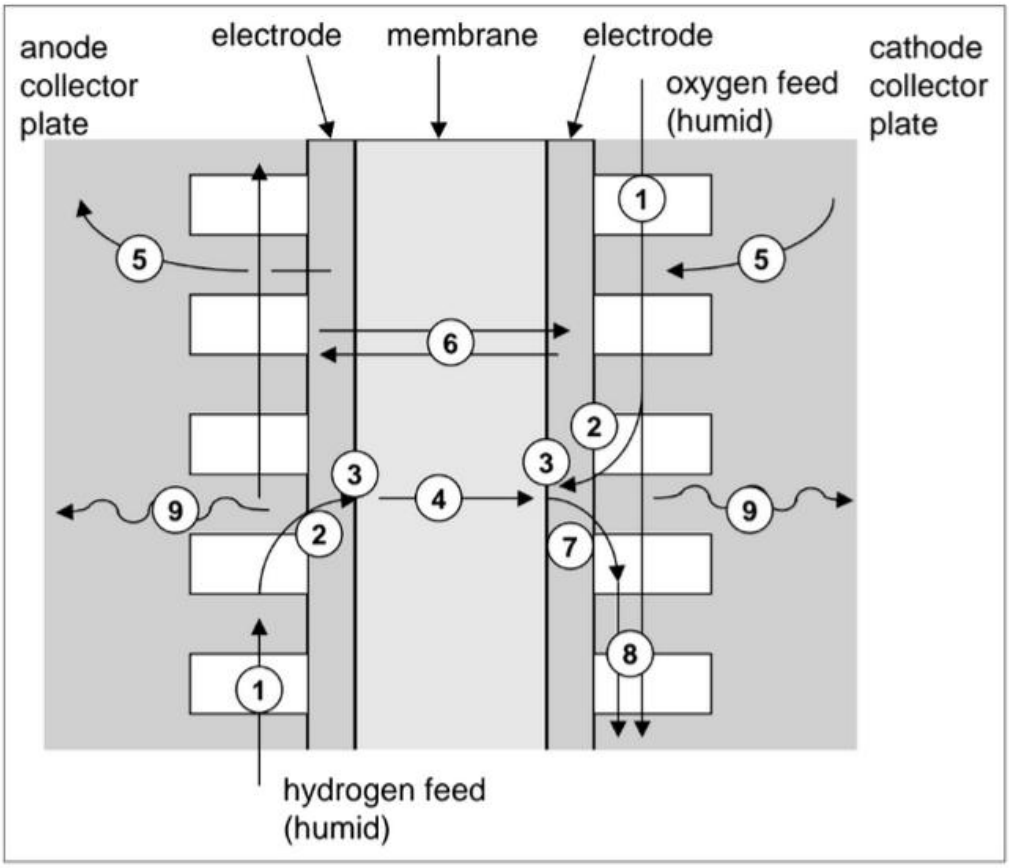


Figure I.6 Main cell components (not in scale) and processes [14].

The table I.1 presents these fuel cell operating processes in a simplified manner.

Table I.1 Eight simultaneous processes in a fuel cell (adapted from Barbir [14]).

	Process	What happens	Where it happens
1	Gas input	Hydrogen and oxygen enter the cell through channels in the outer plates. The gases are fed from one side at controlled pressure and flow rate.	Flow channels in bipolar plates.
2	Gas diffusion	The gases spread out and move through the porous electrodes. This allows them to reach many catalyst sites, not just one spot.	Porous gas diffusion layer.
3	Electrochemical reactions	At the catalyst particles, hydrogen molecules break apart into protons and electrons. At the same time, oxygen reacts with electrons and protons to form water.	Catalyst layer.
4	Proton transport	The positive ions (protons) created from hydrogen move through the polymer membrane. They are pulled by the electric field created by the separation of charges.	Polymer membrane.
5	Electron flow	The electrons cannot cross the membrane, so they must take the external circuit. This forced path is where the electrical current is collected and can power devices.	External circuit and conductive parts.
6	Water movement	Water is produced at the oxygen side, but moves back across the membrane. This happens because water molecules are carried along with the moving protons.	Across the membrane.
7	Water removal	Some water stays in the electrodes as liquid; some evaporates and leaves with exit gases. The balance between these effects cell performance.	Catalyst layer and gas diffusion layer.
8	Heat management	The electrochemical reactions release heat. This heat spreads through the solid materials and is carried away by a cooling fluid or by the gas streams.	Throughout the cell structure.

I.3.2 Electrochemical energy conversion of a fuel cell

A fuel cell converts the chemical energy stored in hydrogen and oxygen bonds into electrical energy after they enter the cell. Water and heat are also produced as by products. The reaction occurs in two steps, at the anode and the cathode [12, 14].

I.3.2.1 Hydrogen Oxidation Reaction (HOR)

When hydrogen is present in the anode of the cell the first reaction which occurs, it is hydrogen splitting. This produces two electrons that travel through the external circuit, to generate electricity. The hydrogen ions then travel through the electrolyte towards the cathode [12], this reaction is given by the equation I.1. The figure I.7 illustrates the HOR in the anode.

$$H_2 \rightarrow 2\,H^+ + 2\,e^- \quad Eq\ I.1$$

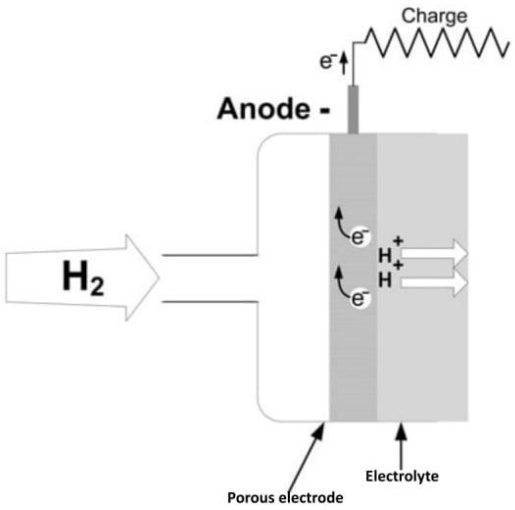


Figure I.7 First step: Hydrogen decomposition at the anode (adopted from [12])

I.3.2.2 Oxygen Reduction Reaction (ORR)

When oxygen (usually air) enters through the cathode channels and then reaches the cathode catalyst layer. The protons that crossed the membrane and the electrons that came back through the external circuit recombine with oxygen to form water [12], the equation I.2 shows all. The figure I.8 illustrates the ORR at the cathode.

$$½\,O_2 + 2\,H^+ + 2\,e^- \rightarrow H_2O \quad Eq\ I.2$$

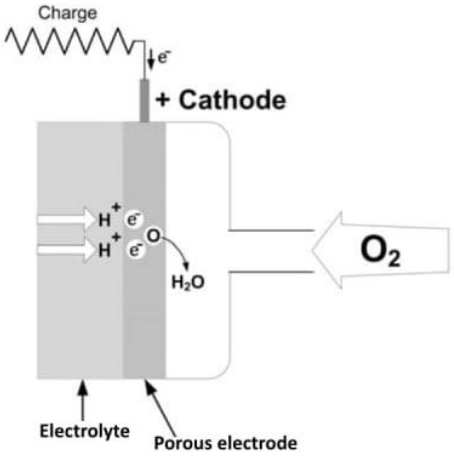


Figure I.8 Second step: Oxygen decomposition at the cathode (adopted from [12]).

I.3.2.3 Overall reaction occurring in a fuel cell

According to IUPAC, when the anode and cathode reactions are combined, water is formed as a liquid according to the I.3 under normal temperature and pressure (0°C and 1 atm) [12]. The figure I.9 represent the electrochemical process within a hydrogen & oxygen fuel cell.

$$2\,H_2 + O_2 \rightarrow 2\,H_2\,0_{liquid} \quad Eq\ I.\ 3$$

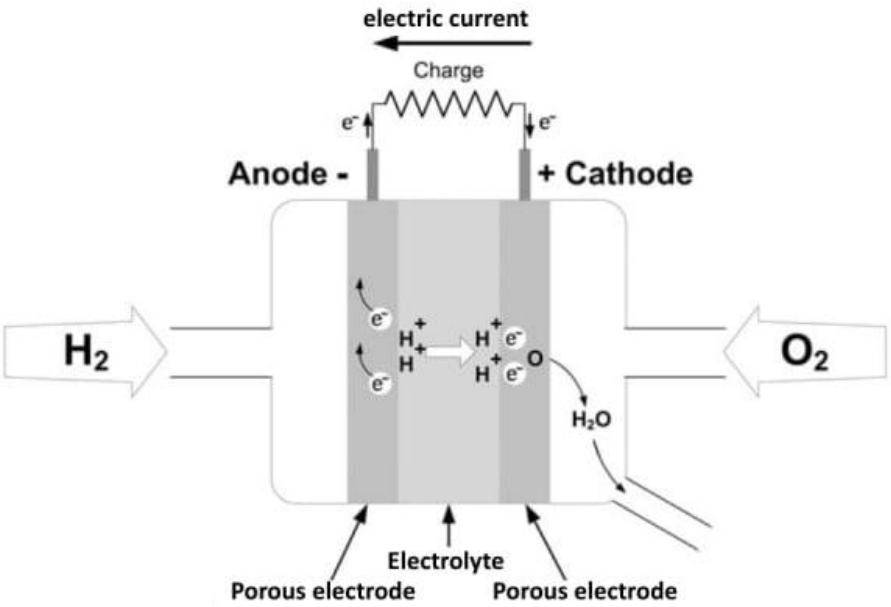


Figure I.9 Overall operating diagram (adopted from [12]).

I.4 Classification of fuel cells

Various families of combustible heaps have been identified. The piles are categorized based on their electrolyte type and operating temperature. The most five types of fuel cell can be distinguished, including:

- Solid oxide fuel cells (SOFCs).
- Molten carbonate fuel cells (MCFCs).
- Proton exchange membrane fuel cells (PEMFCs).
- Phosphoric acid fuel cells (PAFCs).
- Alkaline carbonate fuel cells (AFCs).

I.4.1 Solid oxide fuel cell (SOFC)

The SOFC works at high temperatures, between 600 and 1000 °C. This high temperature is needed because the ceramic electrolyte, made of yttria-stabilized zirconia (YSZ), can conduct ions only when it is hot. At these temperatures the SOFC can use carbon monoxide as fuel. In other types like the PEMFC, CO is a poison that destroys the catalyst, but in the SOFC it is just another fuel [15]. The high heat is also useful: it can be recovered for heating, which gives a global efficiency of 80 to 85 % when used in cogeneration. The electrical efficiency alone is between 50 and 60 %. Another advantage is that the SOFC can use methane directly through internal reforming. For all these reasons, the SOFC is a good choice for natural gas applications [12, 15]. The figure I.10 represents the schematic diagram of SOFC, illustrating the electrochemical process of converting fuel and oxygen into electricity.

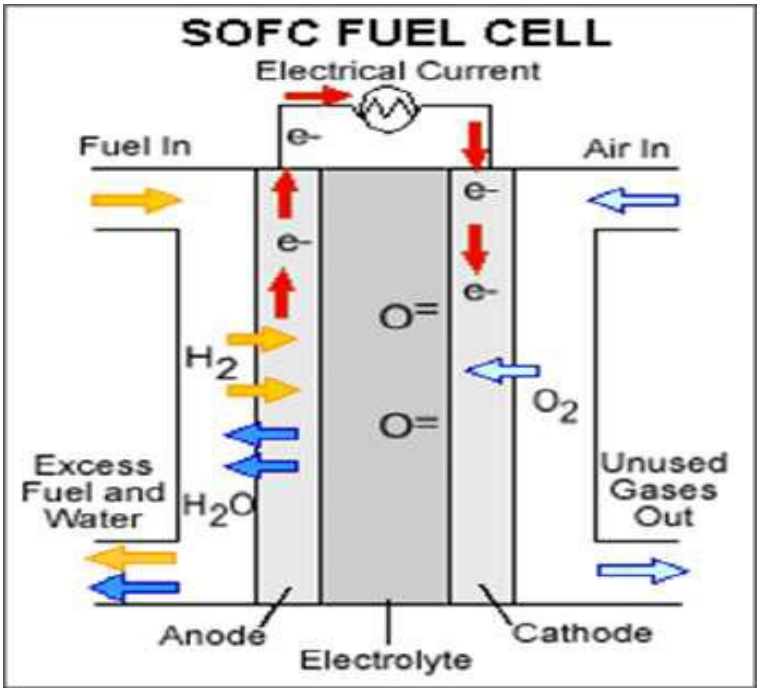


Figure I.10 SOFC schematic diagram [72].

I.4.1.1 SOFC charateristics

The table I.2 summarizes the characteristics and performance of SOFC:

Table I.2 SOFC characteristics [12, 14, 15].

Model	SOFC
Compatible fuels	H_2, CH_4, CO, syngas, biogas, light hydrocarbons
Primary oxidant	O_2 (air)
Electrolyte	YSZ, Solid Ceramic (zirconium dioxide)
Charge carrier	Oxygen ion $\boldsymbol{O^{2-}}$
Anode Material	Ni-YSZ cermet
Cathode Material	LSM ceramic
Operating Temperature	600 – 1000 ºC
Reaction	**Anode:** $H_2 + O^{2-} \rightarrow H_2O + 2\,e^-$
	Cathode: $½\,O_2 + 2\,e^- \rightarrow O^{2-}$
Efficiency	≈60%
Lifespan	> 40 000 hours

I.4.2 Molten carbonate fuel cell (MCFC)

The MCFC works around 650 °C. The electrolyte is a mixture of melted carbonates (Li_2CO_3 and K_2CO_3) held in a porous ceramic matrix [12]. The ion that crosses the electrolyte is the carbonate ion CO_3^{2-}. This creates a problem: the ion is consumed at the anode, so the system has to bring CO_2 from the anode side back to the cathode side. This recycling step makes the MCFC more complicated than the SOFC. Like the SOFC, the MCFC can use methane and tolerates CO. It is mostly used in big stationary plants for industrial cogeneration. The electrical efficiency is around 45 to 55 % [15]. The schematic diagram and the electrochemical process of the MCFC is represented in figure I.11.

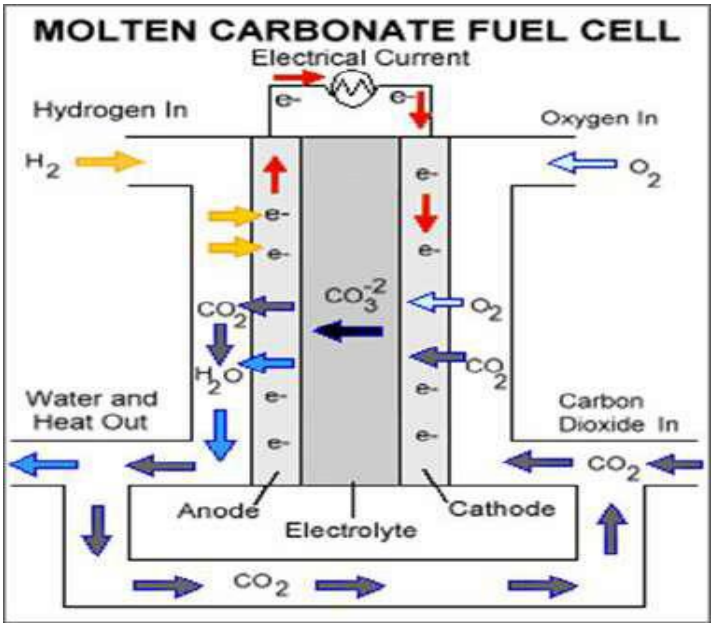


Figure I.11 Schematic diagram of a MCFC [72].

I.4.2.1 MCFC characteristics

The table I.3 represents the characteristics and performance of a molten carbonate fuel cell:

Table I.3 MCFC characteristics [12, 14, 15].

Model	MCFC
Fuel	H_2, CH_4, biomass
Primary oxidant	O_2 , CO_2
Electrolyte	Molten alkali carbonates (Li_2CO_3/K_2CO_3)
Charge carrier	Carbonate ion CO_3^{2-}
Anode material	Nickel
Cathode material	Lithiated nickel oxide (NiO)
Operating Temperature	600 - 700 ºC
Reaction	**Anode:** $H_2 + (CO_3)^{-2} \rightarrow H_2O + CO_2 + 2\,e^-$
	Cathode: $CO_2 + 2\,e^- + ½\,O_2 \rightarrow (CO_3)^{-2}$
Efficiency	≈55%
Lifespan	Several thousand hours

I.4.3 Phosphoric acid fuel cell (PAFC)

The PAFC was the first fuel cell technology that was sold for real commercial use. It works between 150 and 200 °C, with concentrated phosphoric acid (H_3PO_4) as electrolyte [15]. The electrodes use platinum on a carbon support, which is one of the reasons the PAFC stays expensive. The PAFC tolerates around 1 % of CO in the fuel, which is much better than the PEMFC, but it still cannot use methane directly: an external reformer is needed. PAFC units of 100 kW to a few MW are installed in hospitals and hotels, where they produce both electricity and hot water. The figures I.12 shows the schematic diagram and the electrochemical process of PAFC.

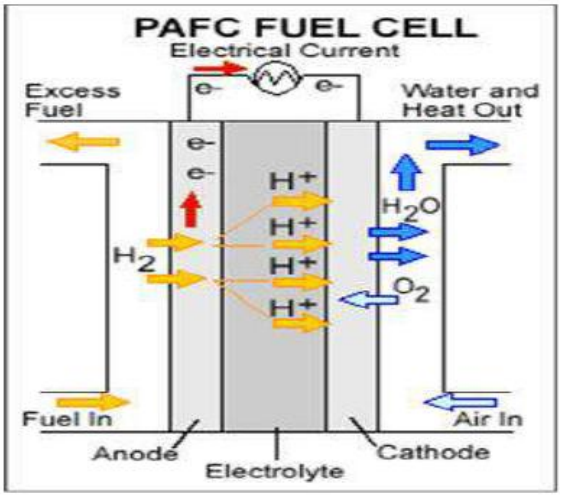


Figure I.12 Schematic diagram of PAFC [72].

I.4.3.1 PAFC characteristics

In the table I.4, we have summarized the various characteristics and performance of phosphoric acid fuel cells:

Table I.4 PAFC characteristics [12, 14, 15].

Model	PAFC
Fuel	H_2
Oxidant	O_2 (air)
Electrolyte	H_3PO_4 (Concentrated phosphoric acid)
Charge carrier	H^+
Anode material	Pt on carbon
Cathode material	Pt on carbon
Operating Temperature	160 – 200 ºC
Reaction	**Anode:** $H_2 \rightarrow 2\,H^+ + 2\,e^-$
	Cathode: $½\,O_2 + 2\,e^- + 2\,H^+ \rightarrow H_2O$
Efficiency	40 - 55%
Lifespan	> 40000 hours

I.4.4 Proton exchange membrane fuel cell (PEMFC)

The PEMFC uses a solid polymer membrane, usually Nafion. The membrane needs to be humid to conduct protons, which is why this type of fuel cell has to work at low temperature, between 50 and 80 °C [8]. The advantages of this low temperature are clear: fast start-up, small size, high power density, and no liquid electrolyte that can leak. But the price is high. The catalyst is platinum, which is expensive, and the cell needs very pure hydrogen. Even 10 ppm of CO is enough to poison the catalyst. For this reason, the PEMFC cannot use methane and is used mostly in transport (cars, buses, forklifts), not in industrial methane recovery [14, 15]. The schematic diagram and the electrochemical of PEMFC are illustrated in the figure I.13.

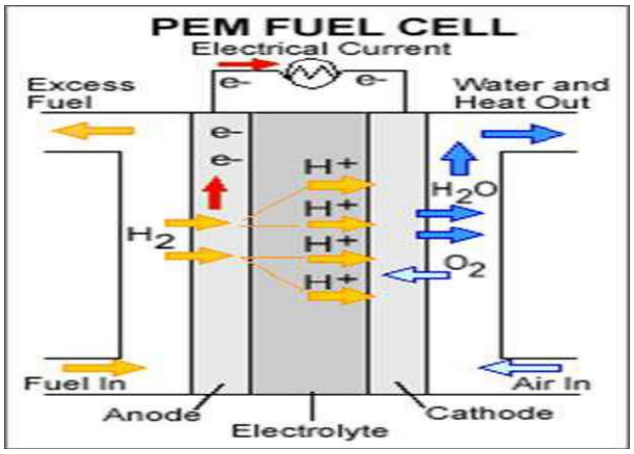


Figure I.13 Schematic diagram of PEMFC [72].

I.4.4.1 PEMFC characteristics

In the table I.5, we have summarized the different characteristics and performance of proton exchange membrane:

Table I.5 PEMFC characteristics [12, 14].

Model	PEMFC
Fuel	H_2
Oxidant	O_2 (air)
Electrolyte	Nafion
Charge carrier	H^+
Anode material	Pt or Ru
Operating Temperature	60 - 80 ºC
Reaction	**Anode:** $H_2 \rightarrow 2\,H^+ + 2\,e^-$
	Cathode: $½\,O_2 + 2\,e^- + 2\,H^+ \rightarrow H_2O$
Efficiency	40 - 50%
Lifetime	> 10000 hours

I.4.5 Alkaline fuel cell (AFC)

One of the earliest fuel cells to be utilized to generate actual power is the AFC. It was utilized by NASA to supply astronauts with drinking water and energy during the Apollo missions and the Space Shuttle. It operates at temperatures between 60 and 90 °C and uses potassium hydroxide (KOH) as an electrolyte. KOH reaction with carbon dioxide is the primary issue with AFC. By creating potassium carbonate (K_2CO_3), even a tiny quantity of carbon dioxide in regular air might harm the electrolyte. As a result, pure hydrogen and oxygen are very necessary to to ensure proper functioning of this fuel cell. As a result, rather than being used for natural gas applications, it has mostly been used in space and military submarines [3]. The electrochemical process and the AFC schematic diagram are shown in figure I.14.

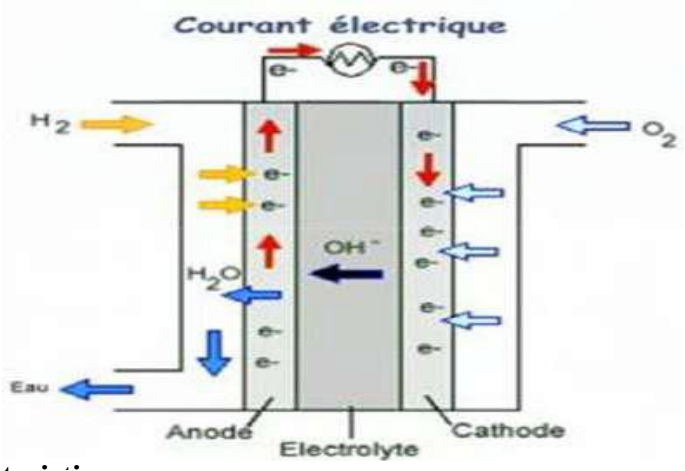


Figure I.14 Schematic diagram of AFC [72].

I.4.5.1 AFC characteris··

The table I.6 represents the characteristics and performance of an Alkaline Fuel Cell:

Table I.6 AFC characteristics [12, 14, 15].

Model	AFC
Fuel	Pure H_2
Primary oxidant	O_2
Electrolyte	KOH
Charge carrier	OH^-
Anode material	Ni, Pt
Cathode material	Ag, Pt
Operating Temperature	60 – 90 °C
Reaction	**Anode:** $H_2 + 2\,OH^- - Ni \rightarrow 2\,H_2O + 2\,e^-$
	Cathode: $½\,O_2 + 2\,e^- + H_2O - Ag \rightarrow 2\,OH^-$
Efficiency	> 60 % (with pure hydrogen)
Lifespan	≈5000 hours

I.5 Advantages and disadvantages of fuel cells

Fuel cells have many benefits. This generator is ideal for the environment because it produces just water and emits no greenhouse gases while the fuel cell efficiently generates electrical and thermal energy in a silent manner. Fuel cells are not yet a fully developed technology, though. Reliability, longevity, cost, and commercial development are just a few of the many areas that still need research and improvement. Each of the numerous fuel cell technologies has pros and cons of its own. The benefits and drawbacks of the various fuel cells are listed in the table I.7.

Table I.7 Table of advantages and disadvantages of different types fuel cells [12].

Type of fuel cell	Advantage	Disadvantages
SOFC	❖ Stable electrolyte ❖ High efficiency in combined heat and power (cogeneration) ❖ High electricity production ❖ Can use fuels other than hydrogen directly	❖ Needs materials that can handle high temperatures ❖ Sensitive to sulfur ❖ Takes a long time to start ❖ Affected by changes in operating temperature ❖ Difficult to manage heat

	❖ Uses low-cost catalysts like nickel or metal oxides	
MCFC	❖ High efficiency ❖ Not sensitive to carbon monoxide (CO) ❖ Use a nickel catalyst ❖ Hydrogen can be produced on-site from hydrocarbons ❖ Suitable for combined heat and power (cogeneration)	❖ Start-up time ❖ Control of the electrolyte (consumption of carbonate ions) ❖ Corrosion of the anode and cathode by the electrolyte ❖ Sensitivity to sulfur ❖ CO_2 management ❖ Loss of electrolyte ❖ Dissolution of the cathode (nickel-based) ❖ Low current density
PEMFC	❖ Very fast start-up ❖ Very fast response time ❖ Compact design ❖ Operates at low temperatures ❖ Not sensitive to carbon dioxide (CO_2) ❖ Fully solid structure	❖ The cost of membrane ❖ Need to keep the membrane humid ❖ Expensive catalyst (platinum) ❖ Sensitive to carbon monoxide (CO) at concentrations above 10-20 ppm and to sulfur ❖ Need to improve lifetime and reliability for real applications
PAFC	❖ Low operating temperature ❖ Not sensitive to carbon dioxide (CO_2) ❖ Slightly sensitive to carbon monoxide (CO),	❖ Start-up time ❖ Cost of the catalyst (platinum) ❖ Degradation due to electrolyte corrosion

	can tolerate up to about 1% ❖ Suitable for combined heat and power (cogeneration)	❖ Need for regeneration with phosphoric acid ❖ Sensitivity to sulfur ❖ Temperature control
AFC	❖ Operates at low temperatures ❖ Works at atmospheric pressure ❖ Low electrolyte cost ❖ Low catalyst cost ❖ High response speed ❖ Fast start-up ❖ High electrical efficiency ❖ Can operate at very low temperatures (below 0 °C)	❖ Affected by carbon dioxide (CO_2) ❖ Has a corrosive electrolyte ❖ Needs very clean/pure gases to operate properly

I.6 Technological choice of SOFC

We chose this particular type of cell because it operates directly on methane gas, which is the aim of our project: to capture and utilize methane emissions from industrial facilities to generate electricity. We will delve into the materials used in its construction and how this fuel cell works with methane gas.

I.6.1 Conventional fuel cells material

I.6.1.1 Electrolyte

The electrolyte mat is used to stabilize the zircone in the YSZ note (for Stabilized Zirconia). The dosing solution in the yttrium may vary between 3% and 12% molar. The compositions of the most frequently used ingredients correspond to a dosage of 3 mol% of yttrium oxide (Y_2O_3) that is resistant to high mechanical resistance to the electrolyte (~1200 MPa in flexion), and 8 mol% of Y_2O_3 that is resistant to high conductivity. temperature (~0.035 W-1 cm-1 à 800°C) [16].

I.6.1.2 Anode

A nickel–yttria-stabilized zirconia cermet, or Ni–YSZ, is frequently employed as the anode with this electrolyte. Nickel is especially useful since it functions as a great catalyst for the electrochemical oxidation of hydrogen. In order to facilitate efficient gas diffusion over a well-connected (percolating) network, the anode usually has a porosity of about 40%. Typically, the nickel component is around 40% by volume, with yttria-stabilized zirconia making up the remaining 60%. At high temperatures (about 650–1200 $S \cdot cm^{-1}$ at 800°C), this composition offers good electronic conductivity and a thermal expansion coefficient that closely matches the electrolyte [16].

I.6.1.3 Cathode

Because of its strong electrocatalytic activity toward the oxygen reduction reaction, LSM (Lanthanum Strontium Manganite) is frequently utilized as the cathode material. It has a porous structure of about 40%, just as the anode, which promotes effective oxygen diffusion across the electrode. A thermally induced tiny polaron hopping process, in which charge carriers shift between lattice sites as the temperature rises, controls the electrical conductivity of LSM. The material reaches conductivity values between 70 and 140 $S \cdot cm^{-1}$ at about 800°C. Under normal SOFC operating circumstances, the conductivity is sufficient even though it drops at lower temperatures [16].

I.6.2 SOFC structure

The defining characteristic of solid oxide fuel cells, after learning that they operate at high temperatures and accept several types of fuel, is that they are divided into two different types, each with its own characteristics and features.

I.6.2.1 Tubular structure

This type comes in the form of a tube with a tight seal at one end, allowing the fuel to circulate inside the tube while the combustible material circulates outside. This is considered the oldest structure and is the origin of the Westinghouse Group [12].

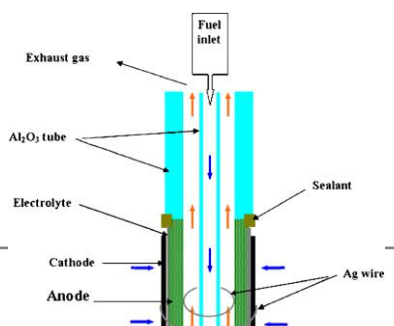


Figure I.15 Tubular SOFC design [73].

I.6.2.2 Planar structure

The difference between the previous type and the second type is the planar structure, which is smaller in size. Since the ionic conductivity of the electrolyte depends primarily on temperature, the thinner the electrolyte, the lower the operating temperature. There are two ways to place the primary cells: the classic structure, which takes the form of plates or sheets, allowing the fuel and oxidizer to reach both sides; and the concentric structure, which has a circular shape, allowing the fuel to reach the center [12].

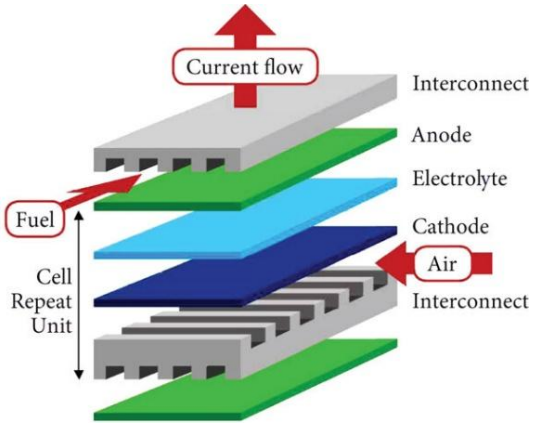


Figure I.16 Palanar SOFC design [17].

I.6.3 Operation of a SOFC under methane

A major advantage of SOFCs is their ability to operate with fuels other than pure hydrogen: ethanol, methanol, or hydrocarbons. Methane is a particularly interesting chemical species in this respect, as it is the major component of natural gas and biogas produced from the treatment of organic waste. This type of fuel can be transformed in a reformer external to the fuel cell into a mixture rich in hydrogen and carbon monoxide. This operating method can be simplified by directly introducing the fuel at the anode, this avoiding the use of an additional reformer. The methane is then converted directly within the anode through the catalytic reactions of steam

reforming (SR) (1.4) and dry reforming (1.5). This is known as Direct Internal Reforming (DIR).

$$CH_4 + H_2O \rightarrow 3\,H_2 + CO \qquad Eq\ I.\ 4$$

$$CH_4 + CO_2 \rightarrow 2\,H_2 + 2\,CO \qquad Eq\ I.\ 5$$

It is important to note that these two chemical reactions are strongly shifted towards the production of H_2 and CO under the operating conditions of a SOFC. Indeed, the nickel contained in the anode is a good catalyst for these reactions, which are also thermodynamically favored at high temperatures (>700°C). The heat absorbed by the endothermic nature of these reactions must then be directly supplied by the heat released from the electrochemical reactions. It should be noted that carbon monoxide, a product of reactions (I.4) and (I.5), can be used to produce hydrogen and carbon dioxide through the water-gas reaction (WGR) (1.6), which is close to thermodynamic equilibrium [16].

$$CO + H_2O \rightarrow CO_2 + H_2 \qquad Eq\ I.\ 6$$

When the (I.4) and (I.6) reactions are combined and assumed to go to completion, the overall reforming is given in the equation I.7:

$$CH_4 + 2\,H_2O \rightarrow CO_2 + 4\,H_2 \qquad Eq\ I.\ 7$$

In our study we assumed that, one molecule of methane produces four molecules of hydrogen if both reactions go to completion. As previously mentioned in the characteristics of SOFC, the reactions at the anode and cathode are modeled based on the overall reforming reaction from (I.7). The reactions become at the anode is given by the equation I.8:

$$4\,H_2 + 4\,O^{2-} \rightarrow 4\,H_2O + 8\,e^- \qquad Eq\ I.\ 8$$

And at the cathode is given by the equation I.9:

$$2\,O_2 + 8\,e^- \rightarrow 4\,O^{2-} \qquad Eq\ I.\ 9$$

By combining these reactions, the overall electrochemical reaction inside the cell is illustrated in the equation I.10:

$$4\,H_2 + 2\,O_2 \rightarrow 4\,H_2O \qquad Eq\ I.\ 10$$

I.7 Electrochemical modeling of a solid oxide fuel cell (SOFC)

The model we use comes from Padullés 2000 [18]. Which is the reference dynamic SOFC model for power systems studies, and from Lakshmi [2]. Who provided the MATLAB/Simulink implementation that we adapted in our work.

I.7.1 Theoretical potential of a fuel cell (Nernst equation)

Every fuel cell has a theoretical maximum voltage that is fixed by thermodynamics. This voltage is also known as the open-circuit voltage since it is achieved when no current is pulled from the cell. The Nernst equation, which links the cell voltage to the standard potential of the reaction, the operating temperature, and the partial pressures of the reactant and product, determines it [15].

For a hydrogen-oxygen fuel cell where the overall reaction is $H_2 + ½\ O_2 \rightarrow H_2O$, the Nernst voltage is written as in equation (2) of Lakshmi [2], and given by the equation I.11.

$$\boldsymbol{E_{Nernst} = E_0 + \frac{RT}{2F}\left(\frac{P_{H_2}P_{O_2}^{0.5}}{P_{H_2O}}\right)} \qquad \textbf{Eq } \textit{I. 11}$$

In this expression, E_0 is the standard potential of the hydrogen-oxygen couple. Its value at 25 °C is 1.229 V for liquid water, and around 1.1 V at typical SOFC temperatures where water is in vapor form, as reported by Lakshmi in their Table I [2]. T is the operating temperature in Kelvin, F = 96485 C/mol is the Faraday constant, and R = 8.314 J/mol·K is the universal gas constant. The number of electrons exchanged per H_2 molecule during the reaction, represented by the **(n)** in the generic Nernst equation [15], and is represented by the **2** in the denominator. P_{H_2}, P_{O_2}, P_{H_2O} are the partial pressures of the respective species, and are given by the equations I.12, I.13, I.14 respectively, and the fuel flow, and oxygen flow rate giver by equations I.15, I.16.

$$\boldsymbol{P_{H_2} = \left(\frac{\frac{1}{K_{H_2}}}{1+\tau_{H_2}}\right)(q_{H_2} - 2K_r I_{fc})} \qquad \textbf{Eq } \textit{I. 12}$$

$$\boldsymbol{P_{O_2} = \left(\frac{\frac{1}{K_{O_2}}}{1+\tau_{O_2}}\right)(q_{O_2} - 2K_r I_{fc})} \qquad \textbf{Eq } \textit{I. 13}$$

$$\boldsymbol{P_{H_2O} = \left(\frac{\frac{1}{K_{H_2O}}}{1+\tau_{H_2O}}\right)(2K_r I_{fc})} \qquad \textbf{Eq } \textit{I. 14}$$

$$\boldsymbol{q_{H_2} = \frac{2K_r}{U_{opt}}\left(\frac{1}{1+\tau_f s}\right)} \qquad \textbf{Eq } \textit{I. 15}$$

$$q_{O_2} = \frac{q_{H_2}}{r_{OH}} \qquad \textit{Eq I. 16}$$

In table I.8, we summarized the parameters of equations and their significations.

Table I.8 Parameters and their significations [2, 19].

Parameters	Signification
q_{H_2}	Fuel flow rate
q_{O_2}	Oxygen flow rate
q_{H_2O}	Water flow rate
N	Number of cells in stack
K_r	Constant $K_r = {}^{N}/_{F}$
K_{H_2}	Molar valve constants of hydrogen
K_{O_2}	Molar valve constants of oxygen
K_{H_2O}	Molar valve constants of water
τ_{H_2}	Response time of hydrogen
τ_{O_2}	Response time of oxygen
τ_{H_2O}	Response time of water
τ_f	Fuel response time in seconds
$Uopt$	Optimum fuel utilization
r_{OH}	Ratio of hydrogen / oxygen

Two physical observations can be drawn from this equation. The Nernst voltage E increases as the partial pressure of hydrogen H_2 or oxygen O_2 rises. Because more reactants provide the reaction a larger driving force, Nernst also rises. Conversely, Le Chatelier's principle states that the Nernst voltage falls as the partial pressure of the product water H_2O rises [15].

I.7.2 Actual cell voltage

As soon as a current starts flowing through the fuel cell, the actual delivered voltage drops below the Nernst voltage. The difference is shared between three types of losses, each one corresponding to a different physical phenomenon [18, 2], and it is given by the equation I.17

$$V_{fc} = E_{nernst} - V_{act} - V_{ohmic} - V_{conc} \qquad \textit{Eq I. 17}$$

The fuel cell's distinctive V-I curve is caused by each of these three losses dominating at a different point in the operating range [15]. The concentration loss V_{conc} predominates close to the maximum current the cell can give, the ohmic loss V_{ohmic} predominates in the medium range, and the activation loss V_{act} predominates at low current. The figure I.17 shows the polarization curve V-I of a solid oxide fuel cell.

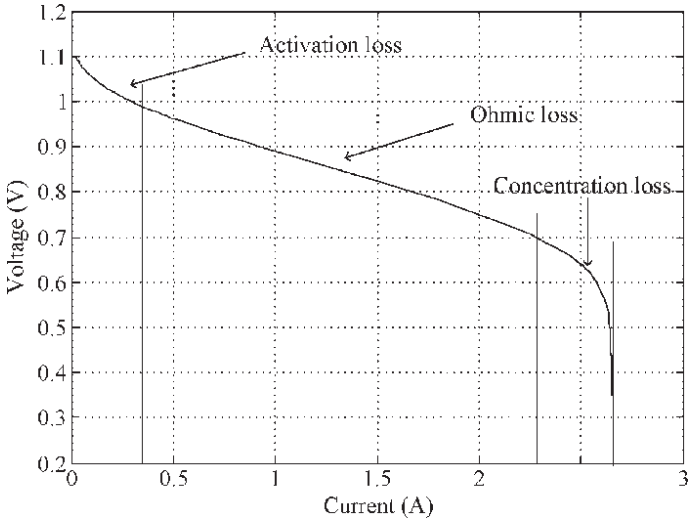


Figure I. 17 Polarization curve (V–I) of a solid oxide fuel cell (SOFC) [20].

I.7.2.1 Activation losses

The activation loss is the voltage that has to be spent simply to start the chemical reactions at the surface of the electrodes. The reaction does not happen for free: there is an energy barrier called the activation energy, and the cell has to push against it [15]. The most common way to model this loss is the Butler-Volmer equation, which is one of the cornerstones of electrochemistry [2]. The equation I.18 describe how to calculate the activation losses.

$$\boldsymbol{V_{act} = \frac{RT}{n\alpha F} ln\left(\frac{I_{fc}}{2I_0} + \sqrt{\left(\frac{I_{fc}}{2I_0}\right)^2 + 1}\right)} \qquad \textit{Eq I. 18}$$

The activation voltage V_{act} depends mainly on two parameters: the exchange current density I_0 and the charge transfer coefficient α. I_0 basically, shows how easy the reaction starts on the electrode surface. When I_0 is higher, the reaction is easier to start, so the activation loss becomes lower. For SOFCs working between 800 and 1000 °C, Gebregergis et al. [20] mention that I_0 is usually around 0.25 mA/cm². The second parameter, α, is the charge transfer coefficient. It gives an idea about how the reaction is shared between forward and reverse directions. In most symmetric reactions, α is taken close to 0.5 [15].

In a fuel cell, the activation loss is the dominant cause of voltage drop at low current. It is responsible for the sharp initial drop seen at the beginning of the V-I characteristic [15]. Once

the current is high enough that the reaction is fully activated, V_{act} increases slowly with current, and the ohmic loss takes over.

I.7.2.2 Ohmic losses

The ohmic loss is the simplest of the three. It comes from the resistance of the electrolyte to ion flow and from the resistance of the electrodes and the interconnects to electron flow. In a SOFC, the electrolyte resistance dominates: even though yttria-stabilized zirconia (YSZ) is a good ionic conductor at high temperature, it is still much more resistive than a metal [15].
The interesting feature of the ohmic loss in a SOFC is that it depends strongly on the operating temperature, following an Arrhenius-type law given by equation (11) of Lakshmi [2]. The equation I.19 we describe how to calculate the internal resistance.

$$\boldsymbol{r(T) = \gamma\, exp\left[\beta\left(\frac{1}{T_0} - \frac{1}{T}\right)\right]} \qquad \textbf{Eq } \textit{I. 19}$$

r(t) is the internal resistance at the operating temperature with γ the pre-exponential factor, is often around 0.2 Ω, the constant related to the activation energy β = −2870 K, T_0 the reference temperature in [K], and T the actual operating temperature of the fuel cell in kelvin according to Lakshmi [2]. The fact that β is negative means that when T increases, the exponent becomes negative and r(T) decreases. Physically, this is because the oxygen ions in YSZ become more mobile as the temperature rises, and the ionic conductivity of the electrolyte improves dramatically [15].

This temperature dependence is one of the main reasons why SOFCs operate at very high temperature, typically between 800 and 1000 °C. Below 600 °C the resistance of YSZ becomes so high that the cell barely produces any useful power [15]. Once r(T) is known, the ohmic loss is simply given by Ohm's law that the equation I.20 illustrates.

$$\boldsymbol{V_{ohmic} = r(T) \cdot I_{fc}} \qquad \textbf{Eq } \textit{I. 20}$$

The ohmic loss is linear in current, which is what gives the V-I curve its almost-straight slope in the middle range of operation [2].

I.7.2.3 Concentration losses

When the fuel cell tries to produce a very high current, a third type of loss appears. At this point, the gases cannot reach the electrodes fast enough. Because of this, the local partial pressures drop, and the reactants near the catalyst are quickly used up. This causes the cell voltage to decrease [15].

The limiting current density I_L is the current level where this problem starts to happen. It depends directly on the fuel flow rate: if the flow rate is higher, more reactant reaches the electrode per second, so I_L becomes higher [2].

Equation (10) in Lakshmi [2] gives the expression for the concentration loss, and it is given by the equation I.21.

$$V_{conc} = -\frac{RT}{nF} ln\left(1 - \frac{I_{fc}}{I_l}\right) \qquad Eq\ I.\ 21$$

I.8 Existing research on fuel cells

Before delving into the technical description of our project, we will review existing research related to fuel cell technology, particularly solid oxide cells.

I.8.1 PEM fuel cell modeling for transportation applications

The first study focused on transportation applications. In 2005, Khan and Iqbal [21] presented a study at a conference where they developed a static and dynamic model of PEMFC. The primary objective of the study was to create an electrical equivalent model of the fuel cell that could be used to simulate the entire power generation system (cell + DC/DC converter) within a vehicle. The results were positive, and the model's validity was confirmed by comparing it with experimental measurements.

I.8.2 Performance Evaluation of Hydrogen Fuel Cell Vehicles

In 2024, authors [22] analyze the performance, energy management strategies, lifecycle cost and emissions of hydrogen fuel cell vehicles. The authors looked at multiple powertrains and compared fuel cell vehicles with conventional internal combustion vehicles. Their main result is that hydrogen fuel cell vehicles consume between 29 % and 66 % less energy than conventional vehicles and emit between 31 % and 80 % less greenhouse gases over their lifecycle, even if the lifecycle cost is still 1.2 to 12.1 times higher.

I.8.3 Dynamic SOFC modeling for power system simulation

The first study that was published on dynamic modeling of SOFC for integration in power systems in 2000. The authors [18] developed a transfer-function model of a SOFC plant that combines the first-order dynamics of each gas species (H_2, O_2, and H_2O) with its own time constant with the Nernst equation for the open-circuit voltage. CFD (Computational Fluid Dynamics) inside power-system simulation tools is not required because the entire model is

built to be fast enough to function. Padullés main contribution is the dynamic element. Before this article, most SOFC models were either steady-state or comprehensive CFD.

I.8.4 MATLAB/Simulink implementation of a dynamic SOFC model

This paper was published in 2013 [2]. The authors took the Padullés transfer-function approach and built a complete MATLAB/Simulink model of a single-cell SOFC, including all three losses (activation, ohmic, concentration). They also validated their results against published experimental data, which gives the model some credibility.

The most useful part of the paper for me is the figures. They show the actual Simulink block diagrams (Figures 1 to 4), the full set of equations with their numerical parameters in Table I, and the V-I characteristics for different flow rates and operating temperatures (Figures 5 to 12).

I.8.5 Experimental study of internal methane reforming in Biogas fed SOFCs

Saadabadi et al. published an experimental study in 2021 on the internal dry reforming of methane in a SOFC fed with biogas. They tested a planar Ni-GDC (Nickel-Gadolinium Doped Ceria) electrolyte-supported single cell with different CH_4/CO_2 mixture ratios between 0.6 and 1.5. Their motivation is that biogas from wastewater treatment plants is commonly flared, which increases greenhouse gas emissions, and they wanted to show that this biogas can be valorized into electricity through a SOFC instead [23].

I.8.6 Methane rich internal reforming in SOFC stacks: Performance evaluation

In 2025 Lin et al. published a recent study [4] on the cell-to-cell variations within methane rich internal reforming SOFC stacks. While most previous research focuses on a single SOFC cell, real industrial systems use stacks of dozens of cells in series, and the cells do not all behave the same way. The authors investigated how the operating conditions vary from one cell to another inside the stack, and what this means for stack design and lifetime. The authors found that the cells closer to the fuel inlet see higher methane concentrations and operate under more endothermic conditions because the reforming absorbs heat. The cells further down see fuel that is already partially reformed and operate under more electrochemical conditions. This creates a temperature gradient and a power gradient across the stack that has to be managed for the stack to operate reliably over its lifetime [24].

I.9 Conclusion

In this chapter, we discussed fuel cells and studied their most important aspects. Now we have a comprehensive understanding of how they work and interact. In the next chapter, we will discuss digital twin technology, which is one of the most important and advanced technologies in the Fourth Industrial Revolution. We will understand the fundamentals, how to model digital twins, and their related characteristics.

CHAPTER II:

Generalities on digital twin

II.1 Introduction

In the last two decades, the industry has moved from a classical vision of production, where the physical and the digital sides of a system were more or less separated, to a much tighter integration between the two. This movement is usually called the fourth industrial revolution, or Industry 4.0, and the digital twin is one of its most emblematic technologies [3, 4]. The idea is simple to explain but not always simple to build: a digital twin is a virtual copy of a real system that evolves with it in real time, thanks to a continuous exchange of data between the physical asset and its numerical model [25].

In the context of our project, which aims to develop a smart digital twin prototype of a Solid Oxide Fuel Cell (SOFC), this chapter is used to give the general background on the digital twin concept before we focus on the fuel cell itself in the next chapter. We start by defining the digital twin and by explaining the differences between a simple model, a simulation and a real twin. Then we describe the architecture of a typical digital twin system, with its physical part, its virtual part and the communication layer. Finally, we present the main technologies that are directly linked to our prototype: the Internet of Things, MATLAB/Simulink as modeling environment, and the MQTT protocol that is used to transfer connect between the real and virtual world.

II.2 History and origin of digital twin

Although the expression 'digital twin' looks recent, the idea of pairing a physical system with a virtual one is not new. The Apollo 13 mission in 1970 is often mentioned as an early precursor: when the explosion occurred on the spacecraft, the engineers on the ground used physical simulators and models of the mission on earth to find a solution and to bring the astronauts back safely [26]. These simulators were obviously not digital in the modern sense, but the logic was already there: reproducing the state of the real system in another environment to understand it and to take decisions.

The term 'digital twin' itself was coined by Michael Grieves in 2003, during a presentation on Product Lifecycle Management at the University of Michigan [27]. At that time, the concept was called 'Mirrored Spaces Model' or 'Information Mirroring Model', and the name 'digital twin' only appeared later in the literature, especially after the NASA paper by Glaessgen and Stargel in 2012. In this paper, NASA and the U.S. Air Force proposed the digital twin as a paradigm for the future vehicles, with the objective of integrating high-fidelity simulation with real sensor data to predict the behavior of the vehicle during its mission [28].

After 2015, the digital twin started to be widely mentioned in the industrial literature, mainly because of the rise of Industry 4.0 in Germany [3] and because the cost of sensors, storage and communication decreased enough to make such systems affordable. In 2017, 2018 and 2019, Gartner included the digital twin among the top strategic technology trends of the year, which accelerated its adoption in several industrial sectors [29]. Today, the concept is used in aerospace, automotive, energy, healthcare, manufacturing and smart cities, with very different levels of complexity depending on the application [25, 30].

II.3 Conceptual and technical foundations of the digital twin

II.3.1 Concept and principals of the digital twin

A digital representation of a physical entity that is linked to it during its entire life cycle and that captures its status and behavior in real time or almost real time is known as a "digital twin" [25, 31]. Michael Grieves initially used the term in 2003 at a Product Lifecycle Management conference at the University of Michigan, and it gained further recognition following the release of a white paper in 2014 [27]. A physical thing in real space, a virtual object in digital space, and the linkages that connect the two in both directions through data and information flows are the three fundamental components of the original concept of the digital twin [27, 32].

The first principle of the digital twin is therefore the idea of a bidirectional link between the physical world and its digital counterpart. Data coming from sensors placed on the real system are sent to the virtual model to keep it synchronized, and the results of the simulation or of the analysis can then be sent back to the physical system, for example as commands or as decision support for the operator [25, 33]. The second principle is the continuous update of the model: contrary to a classical simulation that is used only once during the design phase, the digital twin remains active during the operation of the system, which is why it is sometimes presented as a 'living model' [33]. The figure II.1 the bidirectional link between the physical entities and the virtual models.

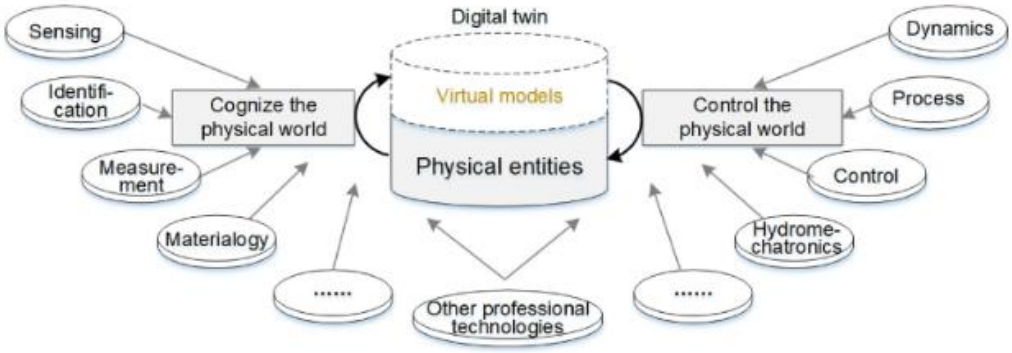


Figure II.1 Enabling technologies for cognizing and controlling physical word [39].

A third principle, which is often added in more recent definitions, is the integration of services on top of the twin. These services can include state monitoring, fault diagnosis, prediction of the remaining useful life, optimization of the operating conditions, or predictive maintenance [25, 30]. Tao et al. [25] even proposed a five-dimensional model of the digital twin (physical entity, virtual entity, services, data, and connections) that is now used as a reference in many academic works.

II.3.2 Types of digital twins

Digital twins are classified into three types:

- Digital Twin Instance (DTI).
- Digital Twin Prototype (DTP).
- Performance Digital Twin (PDT).

II.3.2.1 Digital twin instance (DTI)

A digital twin instance type represents its physical counterpart throughout its whole lifecycle. This type of digital twin allows for continuous monitoring of the physical system state, and any change or development in the real system directly affects its own digital twin and this enables the tracking and prediction of real-world behavior. This type of twin is the best choice in verifying the expected behavior and performance of systems or products [34].

II.3.2.2 Digital twin prototype (DTP)

This type of digital twin relies on collecting and storing data and information related to the physical system. The data may include computer-aided design (CAD) lists, bill of materials (BOM), drawings, or information that connects the manufacturing process to stakeholders in the production chain. Its importance lies in its ability to reduce production costs and uptime by identifying potential defects or risks in the physical twin before production or application. It shares the same characteristics as a digital twin in its ability to simulate manufacturing scenarios and conduct verification and evaluation tests. DTPs can also be called testable DTs, where a virtual prototype becomes available and its level of detail increases gradually. Virtual test results provide a sufficiently reliable statement about the design quality and reduce the number of prototypes required for typically expensive devices. This type of approach is increasingly common in manufacturing and product production processes [34].

II.3.2.3 Performance digital twin (PDT)

Performance digital twin can monitor, collect, and analyze data from products. By combining intelligent capabilities, it can process the data monitored from the physical twin and create actionable data that can be used to improve design, develop maintenance strategies, and draw conclusions from product performance. In real-world situations where we cannot predict the behavior of physical twins, we turn to this type [34].

II.3.3 Integration levels of digital twin

The terms "digital model," "digital shadow," "simulation," and "digital twin" are frequently used interchangeably in the literature. They are not. Based on the degree of automation of the data interchange between the physical and virtual systems [35, 34].

In table I.1, we will give the most popular classification of levels of digital twin.

Table II.1 levels Classification of digital representations [35, 34].

Category	Physical to Digital	Digital to Physical	Example
Digital Model	Manual	Manual	CAD model, standalone Simulink model
Digital Shadow	Automatic	Manual	Real-Time monitoring dashboard
Digital Twin	Automatic	Automatic	Closed-loop predictive control system

II.3.3.1 Digital Model

In a digital model, there is no automatic exchange of data between the real object and its virtual representation: the engineer updates the model by hand when needed, for example during the design phase or when a parameter change. This is typically what is done with a classical CAD model or with an independent Simulink simulation [35, 34]. The data flow of digital model is given in the figure II.2.

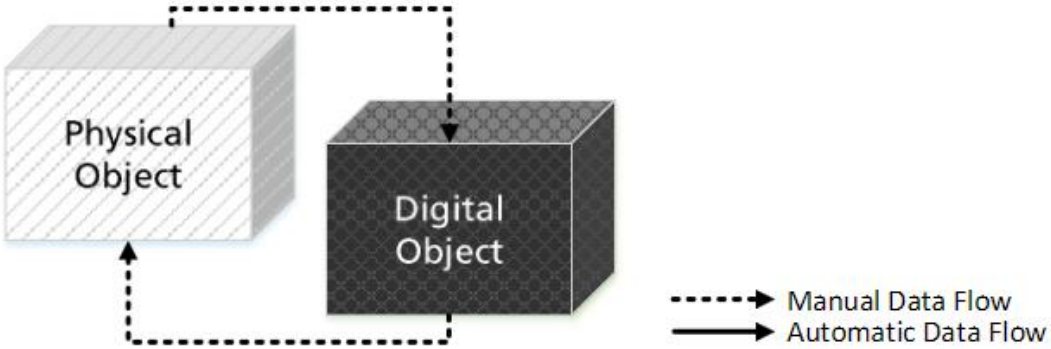


Figure II.2 Data flow in digital model [35].

II.3.3.2 Digital shadow

In a digital shadow, the data go automatically from the physical system to the virtual one, but the opposite direction (from digital to physical) is still manual: the shadow observes the system but does not act on it [35]. Most monitoring platforms used today in the industry are in fact digital shadows, even if they are marketed as digital twins. The figure II.3 illustrates the data flow in digital shadow.

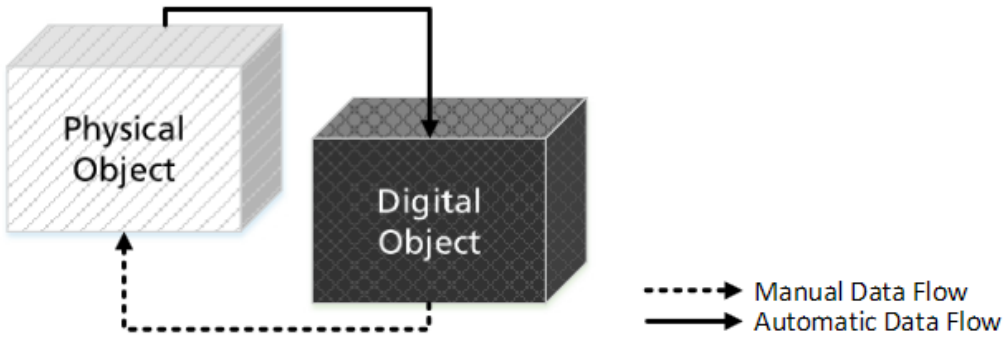


Figure II.3 Data flow in a digital shadow [35].

II.3.3.3 Digital Twin

The real digital twin, involves an automatic two-way exchange: the virtual model receives sensor data continuously and can also send commands in other side or optimized setpoints back to the physical system, and closing the loop [25, 35, 34]. The figure II.4 shows the data flow in digital twin.

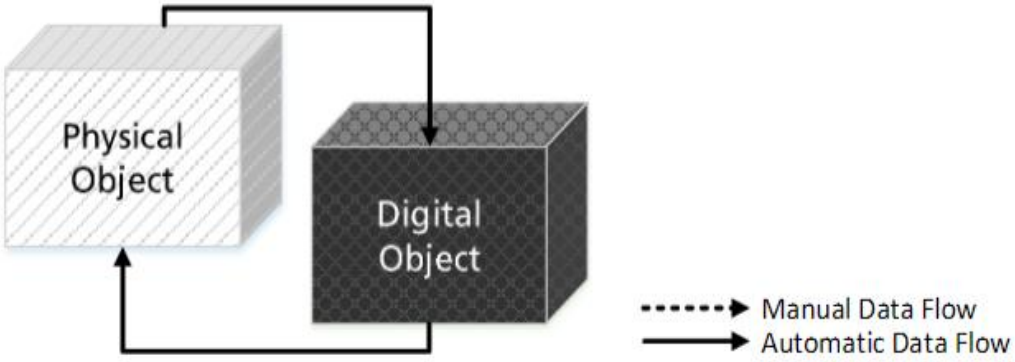


Figure II.4 Data flow in a digital twin [35].

According to this classification, our first prototype is closer to a digital twin, because the ESP32 sends the methane concentration to MATLAB/Simulink automatically through MQTT and send commands back to the physical side.

II.3.4 Architecture of digital twin

A digital twin system is made of a physical component, virtual component, a communication layer that connects between them which is very important, and a real-time synchronization.

The figure II.4 shows the architecture of the digital twin technology.

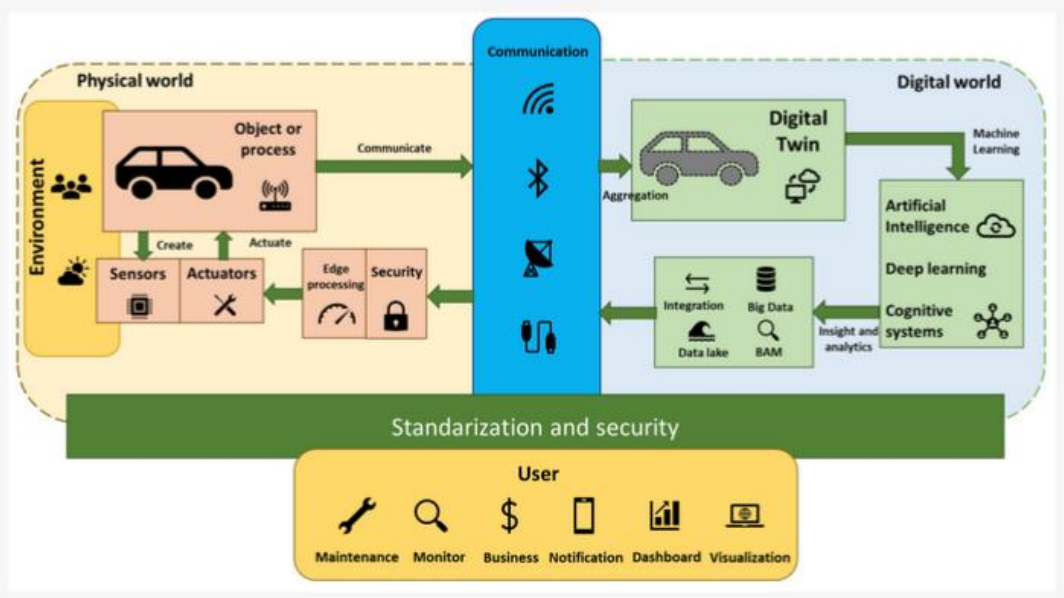


Figure II.5 Digital Twin architecture [34].

II.3.4.1 Physical component

The physical component is the real system whose behavior we want to reproduce in the digital space. This can be almost anything: a machine, a vehicle, a building, or a production line. The

physical component is instrumented with sensors that continuously measure relevant physical quantities such as temperature, pressure, voltage, current, mass flow, or gas concentration, depending on the application [25, 33].

II.3.4.2 Virtual component

The virtual component is the digital representation of the system, and its construction relies on several different methods, including physics-based models that depend on differential equations, empirical models based on data (the latter relying on machine learning), hybrid models that combine the two, Finite Element Models (FEM), or computational fluid dynamics simulation (CFD). All of this depends on the specific details or results that need to be determined [25].

II.3.4.3 Communication system (IoT & Sensors)

To ensure synchronization and facilitate precise simulation and analysis in Digital Twin applications, communication between the physical and digital systems is crucial. The virtual model receives real-time data from sensors and industrial devices and uses it to monitor, forecast, and optimize system behavior [36].

However, dependable communication methods are required to guarantee this ongoing data interchange in industrial settings. High-availability messaging systems are crucial for ensuring steady and continuous data flow between dispersed components, even in complicated or large-scale systems, according to recent research on industrial IoT communication models [37].

II.3.4.4 Real-Time synchronization and data flow

Real-time synchronization is what really distinguishes a digital twin from a classical simulation. In a classical simulation, the model runs in its own time, which has no relation with the time of the physical system. In a digital twin, the model has to follow the physical system at a pace that is compatible with the dynamics of the process: a fast process needs a high sampling rate, while a slow process can tolerate a lower one [25, 33].

The data flow inside a digital twin usually follows the same general architecture: the sensors acquire the raw signal at a given frequency, then microcontroller or gateway performs basic pre-processing (filtering, scaling, and calibration), the data are transmitted to the virtual side through the communication protocol, the model updates its internal state and produces outputs, and finally the outputs are visualized, stored or used to generate a feedback signal [25].

II.4 Industry 4.0 and digital transformation

The digital twin is often presented as one of the pillars of Industry 4.0, the strategic initiative first formulated in Germany in 2013 by Kagermann, Wahlster and Helbig [3]. Industry 4.0 is based on the idea that the new generation of industrial systems should be cyber-physical, meaning that the production equipment is connected, instrumented and able to exchange information with other systems and with the cloud. In this vision, the physical world and the digital world are no longer two separated universes but two sides of the same production system [38].

The 9 technologies usually associated with Industry 4.0 are: the Internet of Things, big data and analytics, cloud computing, additive manufacturing, augmented reality, autonomous robots, simulation, horizontal and vertical system integration, and cybersecurity [38]. The digital twin is strongly connected to several of them, in particular simulation, IoT, cloud and big data, because it needs all of them to operate correctly: sensors to acquire the data, networks to transmit them, cloud or edge platforms to store them, and simulation tools to process them [25].

Beyond the pure manufacturing context, the digital twin is also a key element of the wider digital transformation that is affecting almost all sectors: energy, transportation, healthcare, and smart cities [30].

II.5 Benefits and applications of digital twin

The main benefit of a digital twin is that it allows engineers to observe and analyze the behavior of a system without interacting directly with the physical asset, which is especially valuable for systems that are very expensive, dangerous, remote, or still in the design phase. By running scenarios on the virtual side, it is possible to test the modifications, and anticipate failures, also to optimize the operating conditions before applying them in the real system. This saves time, reduces costs, and limits the risk of damaging the physical equipment [30].

The second main benefit is predictive maintenance. Instead of performing maintenance at fixed intervals or only after a failure, the digital twin is able to monitor the health of the system in real time and to predict when a component is likely to fail, based on the current operating conditions and on historical data [39]. This kind of approach has been widely applied in aerospace (NASA, Siemens), automotive (Tesla, BMW), wind energy (GE, Siemens Gamesa) and more recently in fuel cell systems [5, 40].

II.6 Associated technologies in digital twin

The basic concepts of the most important technologies related to our project and the technologies of digital twin, including the Internet of Things (IoT) as a model for connecting physical and virtual objects, MATLAB/Simulink as an environment for modeling and simulation, and protocols as a communication layer between the controller and the model.

II.6.1 Internet of Things (IoT)

The expression 'Internet of Things' was first introduced by Kevin Ashton in 1999, to describe a vision of the world where every physical object is connected to the Internet and can exchange information with other objects and with humans [41]. More than twenty years later, this vision has largely become a reality: the number of connected devices exceeded the number of human beings on earth around 2010 and keeps growing, with an estimated thirty billion devices connected by 2025 [42].

An IoT system is generally built on four layers: the perception layer, with the sensors and actuators that interface with the physical world; the network layer, which transports the data through wired or wireless communication; the middleware or processing layer, which handles data storage, processing and analytics; and the application layer, where the services offered to the end user are developed [42, 43]. A digital twin can be seen as a specific kind of IoT application, where the perception layer provides the sensor data, the network layer transports them, the processing layer hosts the model, and the application layer presents the information to the operator [25].

II.6.2 Simulation engines

Simulation engines serve as the "brains" of digital twin’s technology. By converting static 3D models into a dynamic, and live replicas. These engines replicate the physical behaviors of the system, and maximize performance throughout product lifecycles by utilizing real-time IoT data and physics-based modeling. These are some examples of software’s that we can use it:

II.6.2.1 ANSYS Twin Builder

A complete software platform called Ansys Twin Builder is used to create, verify, and implement 2D digital models that are driven by hybrid analytics and physics. By enabling engineers to build linked virtual replicas of operational physical assets, maintenance costs can be drastically decreased, equipment downtime can be avoided, and product performance can be enhanced over the course of its lifecycle [44].

II.6.2.2 Simcenter (Siemens)

A key part of the Siemens Xcelerator platform, Simcenter serves as a full suite of simulation, testing, and artificial intelligence tools for creating and managing digital twins throughout the whole product lifecycle [45].

II.6.2.3 MATLAB/Simulink

MATLAB and Simulink are the key associated technologies to digital twin because through them we can do a mathematical modeling of the system, simulation, and control system design in both academic and industrial contexts.

II.6.2.3.1 Choice of MATLAB/Simulink for modeling

MATLAB provides a high-level programming language with a very complete library of numerical and symbolic functions, and Simulink adds a block-diagram environment where dynamic systems can be built, simulated and analyzed in a visual way. MATLAB/Simulink is an obvious choice for the virtual side of a digital twin due to a number of advantages. First, the majority of the requirements of a typical digital twin application are met by the toolboxes devoted to control, signal processing, optimization, and machine learning [46].

Second, Simulink makes it simpler to close the loop between the physical and virtual sides by supporting real-time simulation with the Simulink Real-Time toolbox and communicating with external devices using a variety of protocols (UDP, TCP/IP, serial, and MQTT since R2022a). Third, MATLAB can integrate code written in C, C++, Python or Java, which gives a lot of flexibility when the virtual model is complex or when the communication layer is specific. Several recent works have used MATLAB/Simulink to build digital twins of energy systems [47].

II.6.3 MQTT communication protocol

MQTT (Message Queuing Telemetry Transport) is a lightweight publish/subscribe messaging protocol originally designed by IBM in 1999 for industrial telemetry applications, and standardized by OASIS in 2014 [30]. It was developed with the idea of supporting constrained devices and unreliable networks, which made it a natural candidate for IoT applications when this field started to grow around 2010. Today, MQTT is one of the most widely used protocols for IoT and for industrial digital twin applications [48, 49].

II.6.3.1 MQTT architecture (broker, publisher, and subscriber)

The MQTT architecture is based on a publish/subscribe model with three main actors: the publishers, the subscribers, and the broker [50]. Contrary to a classical client/server model where the client asks directly for information to the server, in MQTT the publishers and the subscribers do not communicate with each other directly. Instead, they both connect to a central broker, which receives the messages from the publishers and forwards them to the subscribers that are interested. The figure II.6 illustrates the architecture of how the MQTT work.

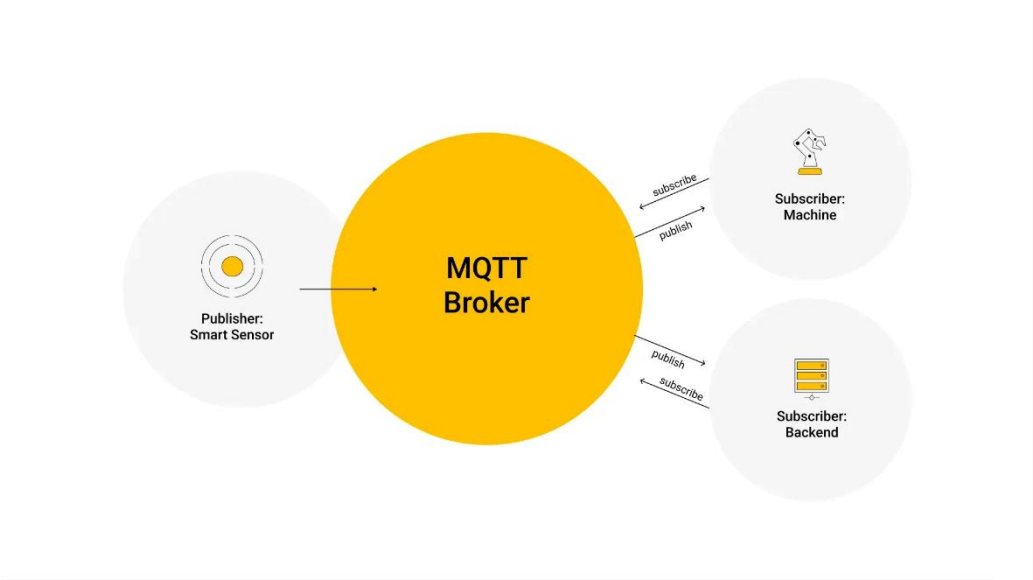


Figure II.6 MQTT Publish/Subscribe Architecture [49].

The exchange is organized around the concept of 'topic'. A topic is a string, hierarchically structured with forward slashes, for example 'AftaFuel/ch4/concentration'. A publisher sends its messages on a given topic, and any subscriber that has subscribed to this topic (or to a matching wildcard pattern) receives a copy of the message [50]. This loose coupling between publishers and subscribers has several advantages: the devices do not need to know each other, they can come and go without breaking the system, and the broker can easily handle hundreds or thousands of clients in parallel.

II.6.3.2 MQTT Qos Levels

One of the features that made MQTT popular in industrial applications is the concept of Quality of Service (QoS). The QoS is a property of each published message and defines the level of guarantee that the message is delivered to the subscribers [50]. Three levels are available, as summarized in the table II.2.

Table II. 2 MQTT Quality of Service levels [50, 51]

Qos	Mechanism	Description
0	At most once	Fire and forgot: A message is sent once and no acknowledgement is required.
1	At least once	Delivered at least once: A message is sent at least once and an acknowledgement is required.
2	Exactly once	Delivered exactly: A four-way handshake mechanism is used to ensure the message is delivered exactly on time.

The choice of the QoS is always a compromise between reliability and network overhead: a higher QoS gives more guarantees but requires more messages on the network and more resources on the client and on the broker [50].

II.6.3.3 MQTT vs Other IoT Protocols

MQTT is not the only protocols available for IoT and digital twin application. The main alternatives are given in the figure II.7.

- HTTP (Hypertext Transfer Protocol) and its secure version HTTPs.
- CoAP (Constrained Application Protocol).
- AMQP (Advanced Message Queuing Protocol).

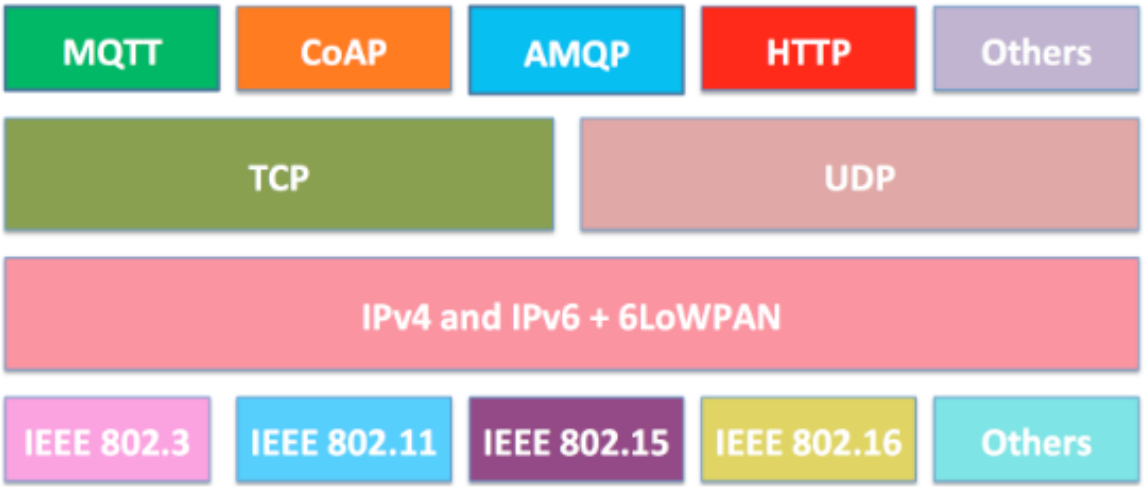


Figure II.7 Protocol Stack for IoT Systems [48].

In the table II.3 we are going to compare and summarize these protocols in terms of many features.

Table II.3 Comparison of MQTT with other IoT protocols [48, 51].

Features	MQTT	HTTP	CoAP	AMQP
Transport	TCP	TCP	UDP	TCP
Architecture	Publish/Subscribe	Request/Response	Request/Response	Publish/Subscribe
Header size	2-bytes	-	4-bytes	8-bytes
QoS	Yes	No	Yes	Yes
Security	TLS/SSL	TLS/SSL	DTLS	TLS/SSL
Application	IoT telemetry	Web services	Constrained devices	Enterprise messaging

In transport there is two types: TCP (Transmission Control Protocol) this type ensures that all data arrives correctly but this requires more time and effort, it is used when data must be complete and dependable. and UDP (User Datagram Protocol) that used when speed and power are more crucial, also to get data rapidly without verifying that everything has arrived.

In security there is also two types: TLS/SSL (Transport Layer Security/Secure Sockets Layer) that secure TCP-based communications, while DTLS (Data Transport Layer Security) provides similar security for UDP-based communications in protocols.

II.7 Existing research on digital twin of fuel cell

Digital twins applied to fuel cells generally aim at three things: tracking degradation, predicting remaining useful life, and improving overall performance. The literature can be split into two main groups depending on the cell technology (PEMFC or SOFC), plus a third group that focuses more on the IoT side of the architecture.

II.7.1 Digital twin for Proton Exchange Membrane Fuel Cells (PEMFC)

Most of the published work is on PEMFC because of the strong demand from the transportation sector. Meraghni et al [52]. were the first to apply the digital twin concept to fuel cells, with a three-layer architecture (physical, connection, digital) and a denoising autoencoder coupled with transfer learning to follow the cell degradation in real time. Yue et al [6, 7]. continued in this direction in 2021 and then in 2022 with a more advanced model combining CNN and LSTM, which gives good results even with small datasets a real advantage when the

system is new and historical data are limite. Zhang et al. [53] also worked on the remaining useful life of PEMFC with LSTM networks and proposed a new loss function called QH-loss to make the online prediction more stable.

II.7.2 Digital twin for Solid Oxide Fuel Cells (SOFC)

Work on digital twins of solid oxide fuel cells is less abundant than for PEMFC, but the topic has attracted more attention in the last few years. Most of the recent papers try to deal with the specific difficulties that come from the high operating temperature of this technology. Testasecca et al. [8] worked on this in 2024. They used Random Forest, XGBoost and some neural networks to predict how efficient a commercial SOFC microcogenerator is, both electrically and thermally. What's interesting is they point out that most papers only care about degradation. Nobody really looks at building a digital twin that works in real time. That's what we're trying to do. Guo et al. [9] did something different they looked at a hybrid system with PV and SOFC together. They used an algorithm called Artificial Rabbits Optimization to figure out the parameters. Tested it on a 5-kW stack. Beloev et al. [10] came more recently, in 2025, and they used old operating data to predict voltage. It's a first step but not a complete digital twin yet.

II.7.3 IoT-Based digital twin for energy system

The combination of IoT and digital twins for energy systems is the direction closest to our own approach. In 2023, a real-time control environment for hybrid energy systems was proposed based on this combination: MATLAB/Simulink was used as the digital counterpart, and several communication protocols such as Modbus TCP/IP, HTTP and PROFINET were used to connect the smart sensors. The application was on photovoltaic systems, but the architecture is very similar to the one we build here for the SOFC [54].

II.8 Research gap and contribution of our work

The Table II.4 summarizes the existing research on digital twin of fuel cells, highlighting the approaches, applications, and limitations of each study.

Table II.4 Summary of existing digital twin research for fuel cells systems [52, 6, 7, 53, 8, 9, 10, 54].

Reference	Year	FC Type	DT Approach	Application	Real-time IoT
Meraghni et al.	2021	PEMFC	Autoencoder + Transfer Learning	RUL prediction	No (historical data)
Yue et al.	2021	PEMFC	Transfer Learning	Early degradation	No (historical data)
Yue et al.	2022	PEMFC	CNN+LSTM+TL	Degradation ID	No (historical data)
Zhang et al.	2023	PEMFC	LSTM+QH-loss	Online RUL	No (short-period data)
Testasecca et al.	2024	SOFC	ML (RF, XGBoost, NN)	Efficiency prediction	No (operational data)
Guo et al.	2023	SOFC+PV	ARO optimization	Parameter ID	No(simulation)
Beloev et al.	2025	SOFC	Data-Driven	Voltage prediction	No (historical operational data)
Chalal et al.	2023	PV	IoT + MATLAB/Simulink	Real-Time control	Yes (but PV, not FC)
Our Work	**2025**	**SOFC**	**Physics-based(padullés) on MATLAB/Simulink**	**Real-Time Control**	**Yes (MQ4+ESP32+MQTT)**

From this review, several observations stand out. The literature on fuel cell digital twins is dominated by PEMFC studies, while SOFC has received far less attention. Moreover, the few existing SOFC digital twin studies mainly rely on historical or simulated data rather than live sensor measurements. To the best of our knowledge, no previous work has connected a real gas sensor to a SOFC digital twin through an IoT communication protocol. Such a connection is beneficial because it allows the digital twin to receive continuously updated methane measurements, react dynamically to real operating conditions, and provide real-time estimation

of fuel cell behavior. This approach improves monitoring capability, enables rapid detection of abnormal conditions, and reduces the need for costly experimental fuel cell setups during the early development stage. The closest related work is the study by Chalal et al., which employed a similar real-time IoT architecture for a photovoltaic system; however, its application to SOFC systems has not yet been reported.

Our work addresses this gap directly and we use the type of DTP because we connect an MQ4 methane sensor to an ESP32 microcontroller, transmit the measurements over MQTT, and use the Padullés 2000 [18] electrochemical model in MATLAB/Simulink as the digital counterpart. The model takes real-time methane concentration as input and computes cell potential, electrical power, temperature, and electrochemical losses.

II.9 Conclusion

In the second chapter, we presented the fundamental and technical concepts related to digital twin technology, highlighting its role in the real-time monitoring, simulation, and optimization of complex systems. In the third chapter, we will present the proposed Smart Fuel Cell Monitoring System, describing its system architecture, hardware components, IoT communication framework, and the operating principle of the developed prototype.

CHAPTER III:

Smart Fuel Cell Monitor Realization.

III.1 Introduction

Sensors, actuators, and microcontrollers may now be integrated into intelligent monitoring and control systems thanks to the quick development of Industry 4.0 and Internet of Things (IoT) technologies. Internet of Things systems depend on networked devices that can gather, send, and process data in real time for analysis and decision-making. Actuators guarantee system reaction and interaction with the environment, while sensors monitor physical factors like temperature, pressure, and gas concentration. Microcontrollers are essential for data gathering, processing, and communication between digital and physical systems [55].

In this chapter, we will present the technical implementation of the proposed IoT-based smart fuel cell monitoring system in detail. We begin by describing the main components that we used in our monitoring system, including gas sensor, ESP32 microcontroller, the buzzer alarm, and the LED indicator, as well as the communication system based on MQTT and the Mosquitto broker. We also present the software tools used for development, including Arduino IDE and the installed broker.

III.2 Overall System Overview:

In our project, we developed a Smart Fuel Cell Monitoring System for methane monitoring and data acquisition. The physical realization of the monitoring system is presented in this chapter, while the Digital Twin environment, including the SOFC Digital Twin Prototype (DTP) model, will be described in detail in the next chapter. Figure III.1 shows the overall setup of the proposed system, including both the physical monitoring and its associated Digital Twin environment.

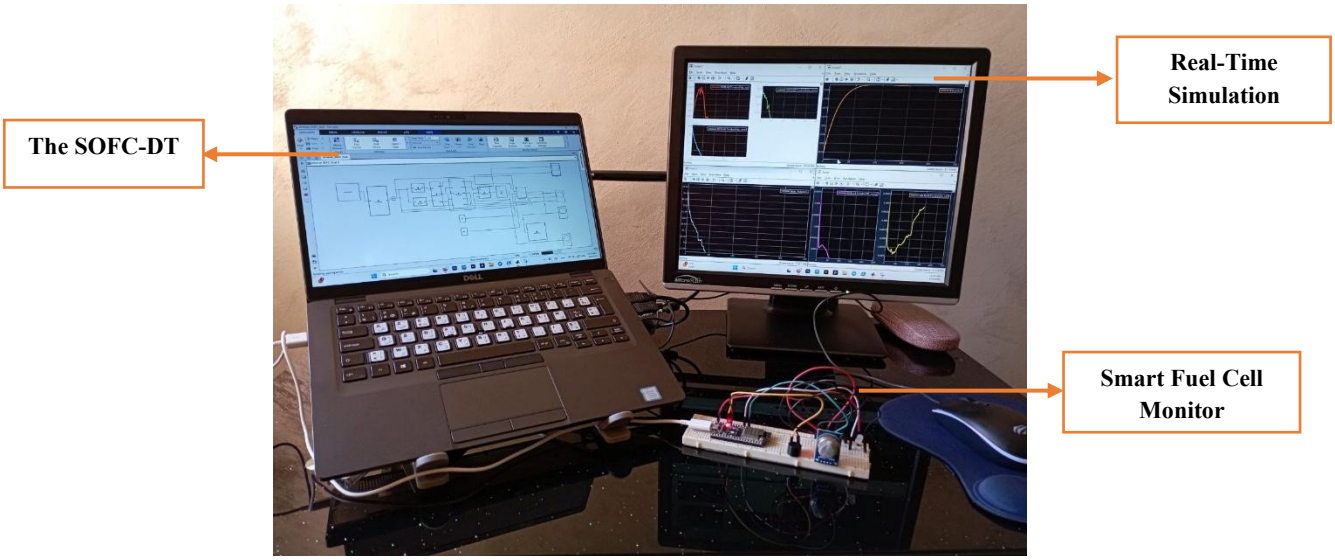


Figure III. 1 Overall implementation of the Proposed Smart Fuel Cell Monitor and SOFC DT.

III.3 Function of Our Smart Fuel Cell Monitoring System

Our monitoring system comprises four essential unit components: Physical sensing unit, processing and communication unit, communication infrastructure, local alert and visualization unit.

Using an MQ-4 gas sensor that detects methane in the environment and converts its concentration into an analog voltage, the proposed monitoring system acquires real-time methane measurements. The ESP32 microcontroller then processes this analog signal, converts it into a methane concentration expressed in parts per million (ppm), and prepares the data for transmission to the SOFC digital twin environment through an MQTT communication network. This enables the digital twin to simulate the behavior of the fuel cell and estimate the potential electrical power that could be generated from the detected methane concentration. In addition, the monitoring system provides real-time feedback through local status indicators, enabling continuous methane detection and system supervision.

By integrating sensing, data processing, wireless communication, and digital simulation, the proposed system enables real-time methane monitoring and supports decision-making. Figure III.2 illustrates the block diagram of the proposed Smart Fuel Cell Monitoring System, highlighting the interaction between the physical sensing unit, the processing and communication unit, the communication infrastructure, and the digital twin environment.

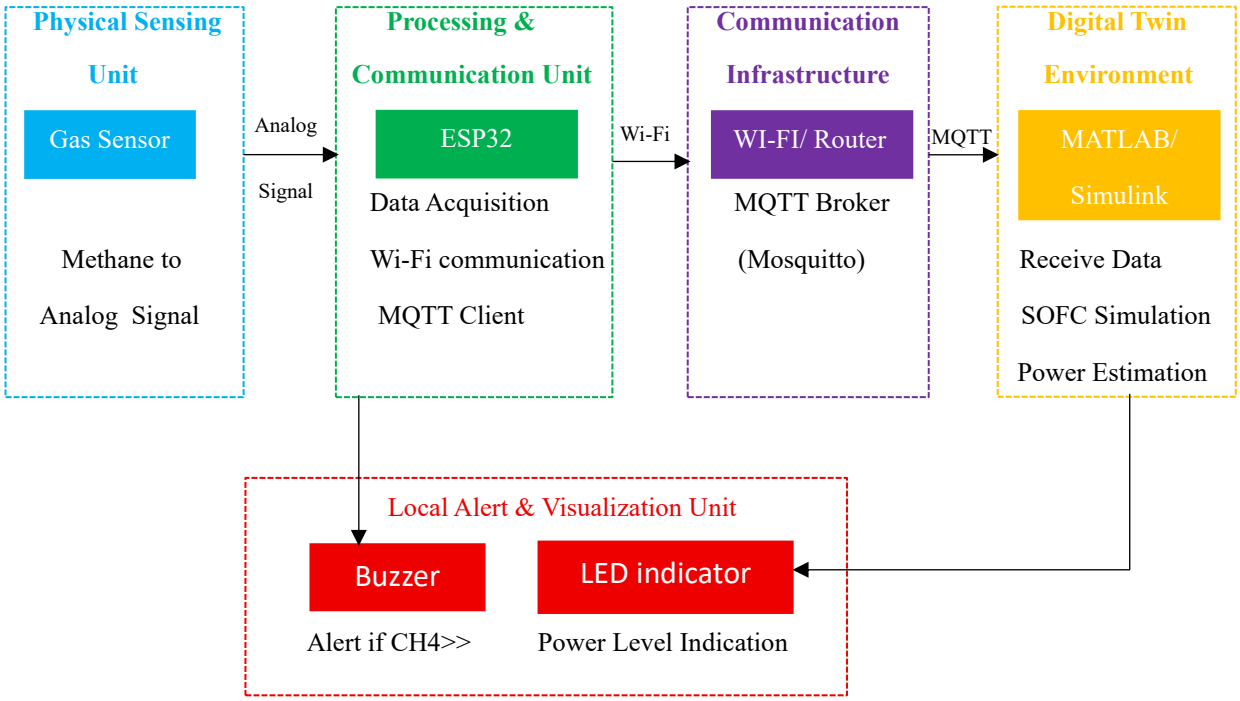


Figure III. 2 Block Diagram of The Smart fuel Cell Monitoring System.

III.4 Hardware and software components used in our monitoring system

The proposed monitoring system relies on the integration of both hardware and software components to enable real-time methane monitoring and communication with the digital twin prototype. The hardware components include the:

- Gas sensor: For methane concentration measurement.
- ESP32 microcontroller: For data acquisition, processing, and wirless communication.
- Buzzer: For alarm activation when methane exceeds.
- LED: Visual indication of the power output estimated by the SOFC model.

And the software components include the:

- Arduino IDE: For programming the ESP32.
- The MQTT protocol and Mosquitto broker: For data exchange between the monitoring system and virtual system.

First, we will explain each component and its basic concepts, starting with the definition of the sensor and the actuators used and how they work in our Smart Fuel Cell Monitoring System, and finally how they are connected to the ESP32 microcontroller.

III.4.1 Gas Sensor

The gas sensor here represents the physical sensing unit that detect the methane gas, then converts it into an analog voltage, and sends this information to the microcontroller. First, we will give the basic concept of a sensor, then we will explain the type of sensor we have chosen to work with, its most important technical specifications, and how we connected it to our microcontroller.

III.4.1.1 MQ-4 Gas sensor

In our project, we use MQ-4 gas sensor, that is designed to detect Methane CH4, and others combustible gases like LPG (liquefied petroleum gas), butane, and propane. Then convert it into an analog voltage. The MQ-4 belongs to the MQ family of gas sensors, which also includes the MQ-2 (smoke), MQ-3 (alcohol), MQ-7 (carbon monoxide) and others [56, 57]. As shown in the figure III.4 the MQ-4 gas sensor used.



Figure III. 3 MQ-4 Gas Sensor.

III.4.1.2 MQ-4 Operating with Methane

Inside the MQ-4 there is a small ceramic tube aluminum oxide (AL2O3) coated with tin dioxide (SnO_2). A heating coil keeps the (SnO_2) layer that maintains it at the proper operating temperature. The resistance is high in clean air because oxygen molecules lie on the tin dioxide's surface and trap free electrons. Methane molecules react with the trapped oxygen when they come into contact with the heated surface, releasing electrons back into the (SnO_2). An analog voltage is measured as the resistance decreases [56]. The components that allow for the measurement of methane gas and then generate an analog voltage are shown in the figure III.4.

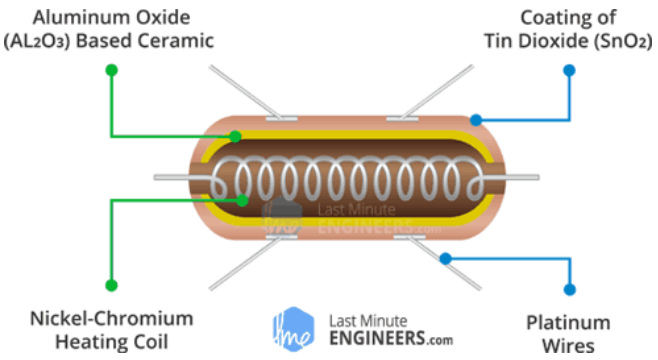


Figure III. 4 Internal construction of a Metal Oxide Semiconductor MQ-series gas sensor [78].

III.4.1.3 The Pins Used in Our MQ-4 Gas Sensor

There are 4 pins in the MQ-4 gas sensor that we used:

- VCC
- GND
- Analog Output (AO): a pin that gives a continuous voltage that is proportional to the gas concentration, we use in our code the AO pin because it gives more precise readings.

- Digital Output (DO): a pin that gives a HIGH or LOW signal depending on a threshold set by a small potentiometer on the board.

The figure III.5 represents the pins located in our MQ-4 gas sensor. These pins allow us to connect this sensor to the ESP32 microcontroller.

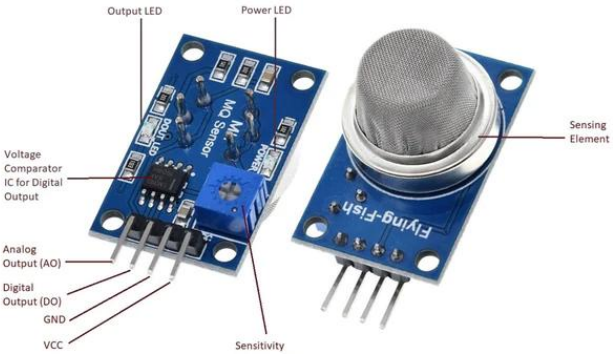


Figure III. 5 Interfacing of MQ-4 Gas Sensor with ESP32.

III.4.1.4 Characteristics of the MQ-4 gas sensor

The Table III.1 summarizes the most important characteristics of a MQ-4 gas sensor.

Table III. 1 Technical characteristics of Our MQ-4 sensor [56, 57].

Property	Value
Target gas	CH_4
Detection range	200 to 10000 ppm
Operating voltage	5V
Heater power	Less than 750 mW
Sensor resistance Rs	10 kΩ to 60 kΩ
Preheat time	24 hours for stable readings
Outputs	Analog (AO) and digital (DO)
Operating temperature	-10 °C to 50 °C
Operating humidity	Less than 95%

III.4.1.5 Wiring of the MQ-4 with The Microcontroller

The MQ-4 sensor needs 5 V to power its heater, so we cannot power it from a GPIO pin of the ESP32. We connect VCC to the Vin power input pin of the ESP32 Devkit, which is directly connected to the 5 V of the USB. The analog output AO is connected to GPIO 32 (an ADC1 input-only pin). The digital output DO is not used. The wiring is given in Table III.2. As also show in figure III.6, which illustrates how to connect this sensor to the microcontroller using online WOKWI simulator software.

Table III. 2 Wiring connections and pin functions for the MQ-4.

MQ-4 pin	NodeMCU-32S	Function

VCC	Vin	Power supply for the heater
GND	GND	Common ground
AO	GPIO 32	Analog Output (gas concentration)
DO	Not connected	Digital Output (not used)

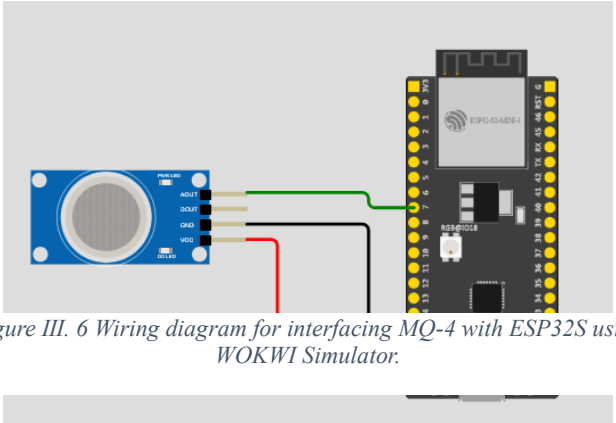


Figure III. 6 Wiring diagram for interfacing MQ-4 with ESP32S using WOKWI Simulator.

III.4.2 ESP32 Microcontroller

The ESP32 Microcontroller represents The Processing and Communication Unit because it acts as an interface between the physical sensing unit and the virtual SOFC model. First, we will give a comprehensive overview of this microcontroller, its types and what we have chosen to work with, the most important concept related to it, and how it works in our Smart Fuel Cell Monitoring System.

III.4.2.1 Origin of the ESP32 Microcontroller:

The Shanghai-based Chinese company Espressif Systems manufactures the ESP32, a low-cost microcontroller. As the replacement for the previous ESP8266, it was introduced in 2016. Because Wi-Fi and Bluetooth are already included into the chip, the ESP32 is a particularly popular microcontroller in Internet of Things (IoT) applications prototype. This eliminates the need for an external communication module. Additionally, it is affordable, compact, and capable of running practical applications like wireless monitoring, data logging, and remote control [58].

III.4.2.2 The ESP32 Microcontroller family

Since 2016, Espressif has released several variants of the ESP32. Each one targets a different need (low cost, low power, more memory, Artificial intelligence integration, etc.). The table III.3 lists the main members of the ESP32 and the variations present in each type.

Table III. 3 Main variants of the ESP32 family [59].

	Variant	Type	CPU	Wireless	Year
Original Series	ESP32	Soc	Dual-Core Xtensa LX6	WiFi + Bluetooth Classic + BLE	2016
The S-Series	ESP32-S2	Soc	Single-Core Xtensa LX7	WiFi only	2019
	ESP32-S3	Soc	Dual-Core Xtensa LX7	WiFi + BLE 5.0	2020
The C-Series	ESP32-C3	Soc	Single-Core RISC-V	WiFi + BLE 5.0	2020
	ESP32-C6	Soc	Single-Core RISC-V	WiFi 6 + BLE + Zigbee/Thread	2022
The H-Series	ESP32-H2	Soc	Single-Core RISC-V	BLE + Zigbee/Thread	2022

III.4.2.3 The differences between ESP32 chips, modules, and development boards

Although the term "ESP32" is frequently used to refer to a microcontroller chip, developers and device manufacturers can work with it in three different formats: chip or Soc (System on chip), module, and development board or Devkit. This is a quick comparison of ESP32 models [60]:

III.4.2.3.1 ESP32 chips

ESP32 chips. The microcontroller, wireless connectivity, and other parts required for the device's functionality are all included in an ESP32 chip, a tiny integrated circuit. The developers must solder it to the module because it is not protected. If you require a more thorough comparison of ESP32 chips, these chips are ideal for unique designs with limited space [60].

III.4.2.3.2 ESP32 modules

The insulated ESP32 chip and additional parts like a voltage regulator, crystal oscillator, and antenna make up an ESP32 module, which is a pre-made, self-contained device. It is a simple way to give IoT devices wireless connectivity and may be put in the MCU [60].

III.4.2.3.3 ESP32 development boards

An ESP32 development board is a whole hardware platform that includes the ESP32 chip and additional components such onboard sensors, USB connections, voltage regulators, and antennas. It's the perfect and ideal method for building and rapidly testing our IoT device. Important features like GPIO and ADC pins, Wi-Fi antennae, LEDs, shielding, and flash memory should be taken into account while selecting the finest ESP32 development board [60].

The figure illustrates the hierarchical levels of the ESP32 hardware integration, showing a raw System on Chip (SoC) is integrated into a shielded Module. These modules are then mounted into a Development Kit (Devkit).

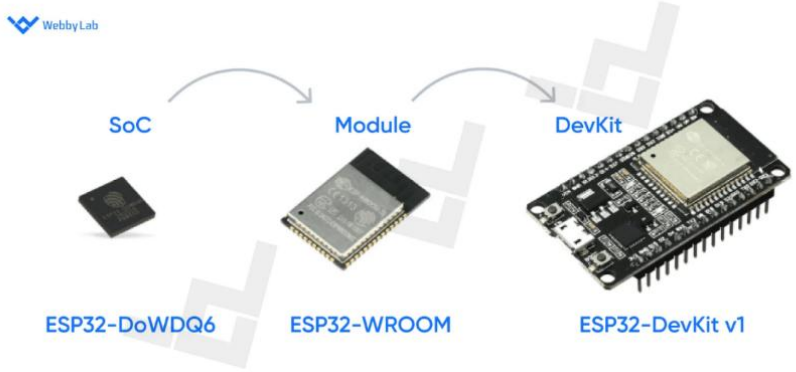


Figure III. 7 Hierarchy of ESP32 Hardware integration [60].

III.4.2.3.4 The comparison between ESP32 chips, modules, and development boards

The ESP32 chips, modules and development bords are the fundamental building blocks in device construction, but each one has its own unique features, characteristics, and application range. The Table III.4 we will compare them in term of Ideal use Cases, Integration, Ease of Use, and Customization.

Table III. 4 Comparaison of ESP32 Hardware formats [60].

Feature	ESP32 chip	ESP32 Module	ESP32 DevKit
Ideal use cases	Projects with little space.	IoT device that require wireless connectivity.	Testing, educational initiatives, and prototyping.
Integration	Unsheilded raw chip.	Shielded, pre-assembled module with essential parts	Completely built, with sensors, USB, and other components.

Ease of use	Advanced (needs a custom PCB (Printed Circuit Board).	Moderate (often plug-and-play)	Simple (ready to use for rapid development).
Customization	High (needs a unique PCB design).	Medium (standard module with integrated parts).	Low (limited modification on a pre-built platform).

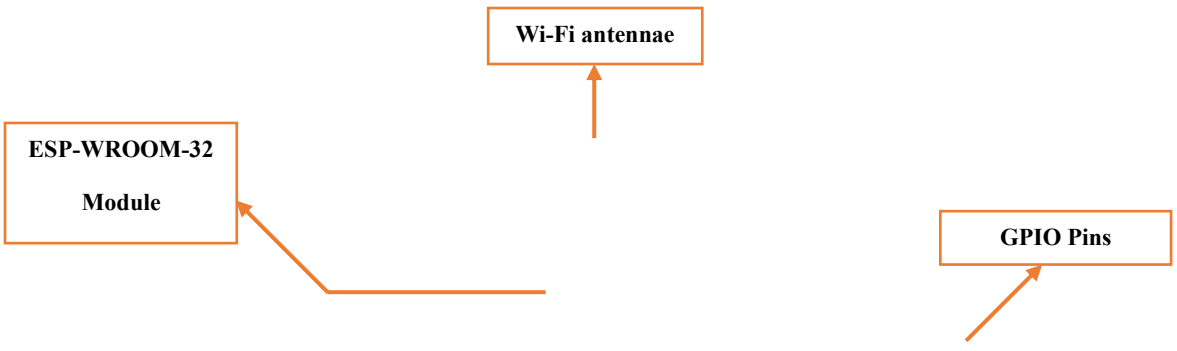


III.4.2.3.5 Type of ESP32 used in our Smart Fuel Cell Monitoring System

In our prototype we utilized the NodeMCU ESP32-S development board (Devkit). It integrates an ESP-32S module (also called the ESP-WROOM-32) because it includes Bluetooth Classic plus BLE and Wi-Fi, for our project. We selected this version for the three reasons:

- It is useful for data transmission because it is compatible with Bluetooth and Wi-Fi.
- The board adds practical electronics, and breaks all the pins out to headers so we can easily plug it into a breadboard.
- It offers the most library support and the biggest community in the Arduino IDE.

The figure III.8 illustrates the key components of our NodeMCU ESP32-S Devkit.



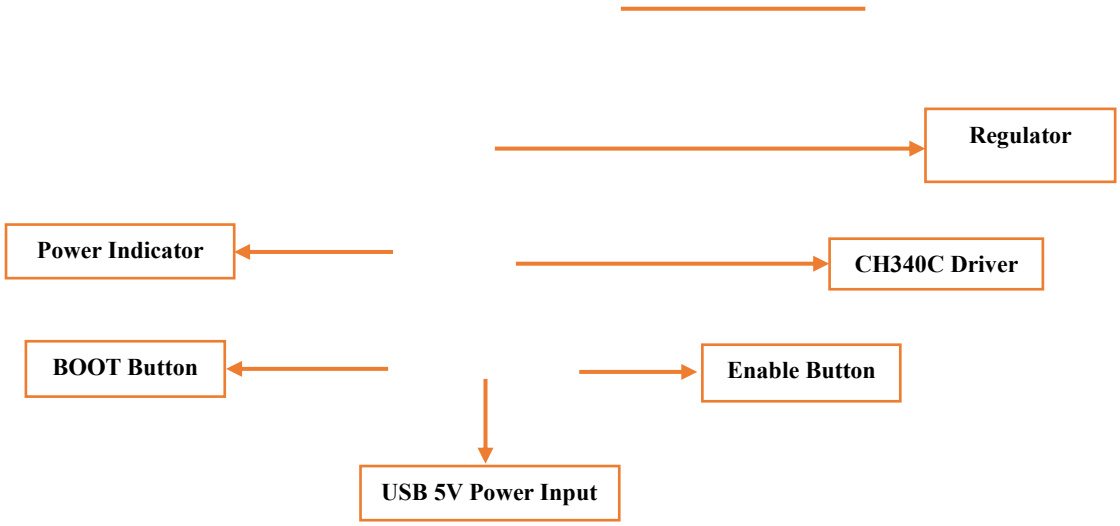


Figure III. 8 NodeMCU ESP32-S development board.

III.4.2.3.6 Function of The NodeMCU ESP32-S in the Monitoring System

The chosen controller in this system carries out seven fundamental tasks, which we shall describe in the sections that follow:

- **Data Acquisition:** Analyzes the analog signal generated by the MQ-4 gas sensor.
- **Data processing:** Determines the methane concentration values in [ppm] using sensor readings.
- **Threshold monitoring:** Compares the recorded concentration to the predefined threshold.
- **Alarm control:** The buzzer emits an audible alert when the concentration of Methane exceeds the threshold.
- **Communication:** The device uses Wi-Fi to transmit methane data to an MQTT interface.
- **Real-time data interchange** between the MATLAB/Simulink virtual SOFC model and the physical sensor equipment is made possible by digital twin integration.
- **Power Visualization:** Uses the virtual model's projected power value to operate the LED.

III.4.2.3.7 Technical Characteristics of the NodeMCU-32S

III.4.2.3.7.1 Specifications

The NodeMCU-32S works well in terms of high integration, wireless transmission distance, power consumption, and network connectivity, and it offers industry-leading parameters. These are some of the specifications we will mention [61]:

- CPU: Xtensa dual-core 32-bit LX6 microprocessor.
- Memory: 520 KiB SRAM.
- 802.11b/g/n (802.11n, speed up to 150 Mbps).
- WIFI Frequency Range 2.4 GHz ~ 2.5 GHZ.
- Clock freaquency adjustment range from 80 MHz to 240 MHz, support for RTOS.
- Support UART, GPIO, ADC, DAC, SDIO, SD card, PWM, I2C, I2S interfaces.
- Encourage secondary development in a combined Windows and Linux development environment.

III.4.2.3.7.2 NodeMCU-32S Pins definitions

There are 38 interfaces in all on the NodeMCU ESP32-S. The figure III.9 shows the various General-Purpose Input/Output (GPIO) pins, analog to digital converter (ADC) channels, communication interfaces (UART, SPI, and I2C), as well the power supply pins.

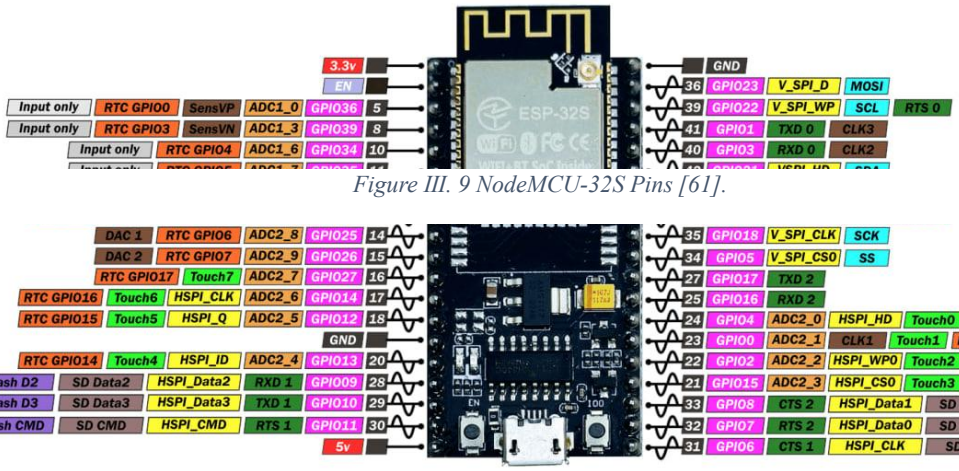


Figure III. 9 NodeMCU-32S Pins [61].

an increase in methane levels, and drives the LED indicator used to represent the power estimated by the virtual SOFC model. The wide range of communication and input/output capabilities offered by the ESP32S makes it the central processing and communication unit of the developed monitoring system.

III.4.3 Buzzer and LED

In our Smart Fuel Cell Monitoring system buzzer and LED represents the Local Alert and Visualization Unit, because it provides real-time feedback on the operating conditions of the

monitoring system. It consists of a buzze
ESI ı we
and cal
mic onit







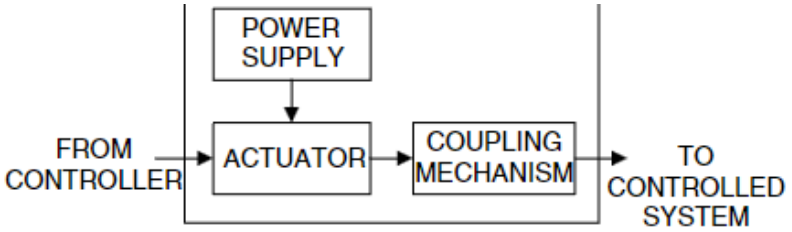
Passive Buzzer Active Buzzer

III.

A r is
action, such as moving, making a sound, li
on, whereas a sensor reads information from the surroundings. It transforms an electrical signal into a physical action after receiving it from the controller [62]. The figure III.10 illustrates what we explained about how actuators work.

Figure III. 11 A typical actuating unit [62].

A buzzer is a n the current passes throug

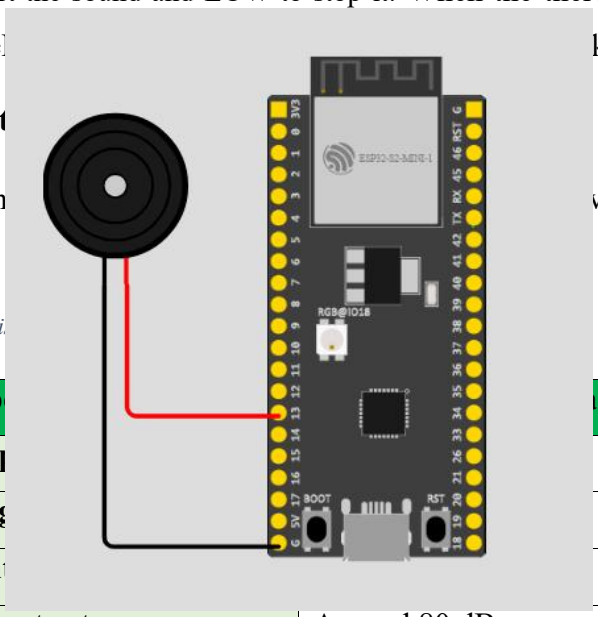


- **Activ** at a specific frequ some type) is suppl) signals.
- **Passive buzzer:** It does not have an oscillator built in. It needs an external square wave signal in order to generate sound. We may play simple melodies using the tone () method, where the pitch is determined by the signal's frequency.

Figure III. 12 Active buzzer.

Figure III. 13 Difference between Active and Passive buzzer.

We chose an active buzzer because we only need a simple beep alarm, not a melody. It is also easier to control: just HIGH to start the sound and LOW to stop it. When the there is an increase in CH4 concentration, the Node kes a beeping sound.



III.4.3.1.1 Characterist

We will summarize th which is an active buzzer in Table III.5.

Table III. 5 Technical characteri

Prop	lue
Ty	
Operating	
Curren	
Sound output	Around 80 dB
Frequency	Around 2.3 KHz
Pins	Positive (+) & negative (-)

III.4.3.1.2 Wiring of the buzzer with ESP32S

In our prototype we connected the buzzer directly to GPIO 13. The wiring is given in Table. And the figure III.6, which illustrates how to connect this active buzzer to the microcontroller using online WOKWI simulator software.

Table III. 6 Wiring connections and pin functions for the Active buzzer.

Buzzer pin	ESP32S pin	Function
(+) long leg	GPIO 13	Control signal from microcontroller
(-) short leg	GND	Common ground

Figure III. 10 Wiring diagram for interfacing Active buzzer with NodeMCU ESP32-S using WOKWI Simulator.

III.4.3.2 Light Emitting diodes (LED)

An LED (Light Emitting Diode) is a semiconductor device that converts electrical energy directly into light; the light comes directly from the recombination of electrons and holes inside a P-N junction. LEDs are used everywhere because they are cheap, small, last for tens of thousands of hours, and consume very little energy [64].

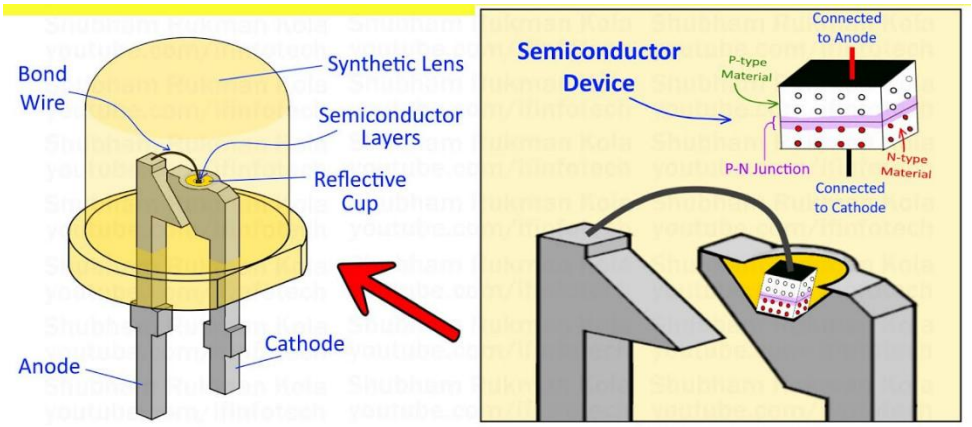


Figure III. 11 Internal structure and components of a light emitting diode.

The LED brightness is controlled according to the power output estimated by the SOFC Digital Twin model in MATLAB/Simulink, providing a visual indication of the simulated electrical power generation.

III.4.3.2.1 Characteristics of the LED

In our project we use one 5mm LED as visual indicator. We will summarize the caracteristics of light emitting diode used in Table III.7.

Table III. 7 Technical characteristics of standard 5 mm LED [65].

Property	Value
Polarity	Anode (+): long leg Cathode (-): Short leg
Forward current	20 mA
Peak forward current	30 mA
Suggestion using current	16 ~ 18 mA
Forward voltage	Min: 1.8 V Max: 2.2 V
Power dissipation	105 mW
Operating temperature	40 ~ 85 °C

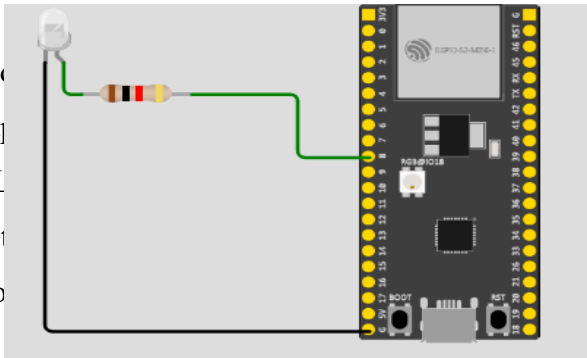
III.4.3.2.2 Wiring of the LED with the ESP32S

The LED is connected to GPIO 13 with a resistor of 220 Ω in series. The wiring is given in table III.8. Also, the figure III.16, which illustrates how to connect this LED to the microcontroller using online WOKWI simulator software.

Table III. 8 Wiring connections and pin functions for the LED.

LED pin	ESP32S pin	Function
(+) long leg	GPIO 25	Control signal from microcontroller
(-) short leg	GND	Common ground

Figure III. 16 Wiring Diagram for Interfacing LED with NodeMCU ESP-32S using WOKWI Simulator Software.



III.4.4 Arduino

In our project we use opment Environment) to write our program and u -source, compatible with Windows, macOS, and L his software IDE can be used with a number of o evant board package has been installed via the Bo

A program written in the Arduino IDE is called a sketch, it includes at least two functionalities, and is saved with the. ino extension. There are two functions must be included in each sketch at least:

- Void setup (): only executes once upon board reset or power-on. Initialization uses pinMode() to set pin modes, start the Serial monitor, establish a Wi-Fi connection, etc.
- Void loop (): continues to run indefinitely. The program's primary logic (reading the sensor, checking the threshold, turning on the buzzer, etc.) is located here.

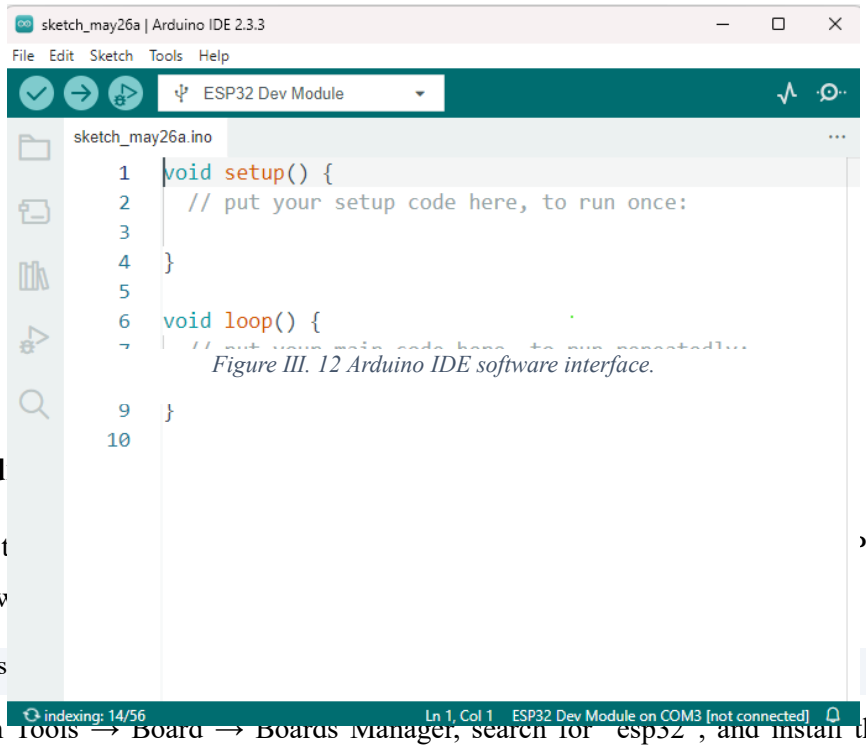


Figure III. 12 Arduino IDE software interface.

III.4.4.1 Add

By default, t ›references and paste the follov

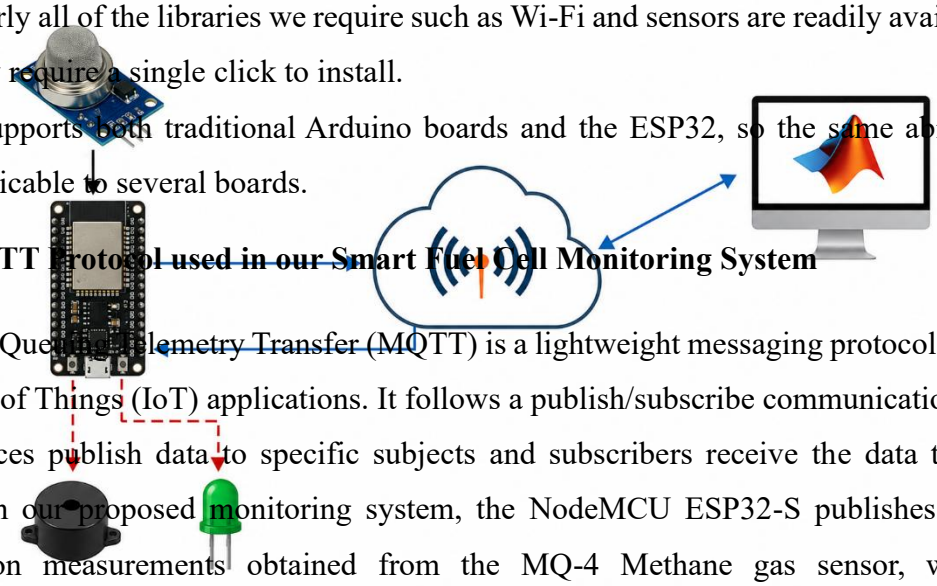
https://dl.espress

Then we open Tools → Board → Boards Manager, search for "esp32", and install the package by Espressif Systems. After that, the ESP32 boards appear in Tools → Board, and we can select "ESP32 Dev Module" and upload our code.

III.4.4.2 Why we chose Arduino IDE

- It is simple to install on any operating system and is free.
- We will benefit from its easy-to-use the interface.

- A sizable online community exchanges tutorials and examples.
- Nearly all of the libraries we require such as Wi-Fi and sensors are readily available and only require a single click to install.
- It supports both traditional Arduino boards and the ESP32, so the same abilities are applicable to several boards.

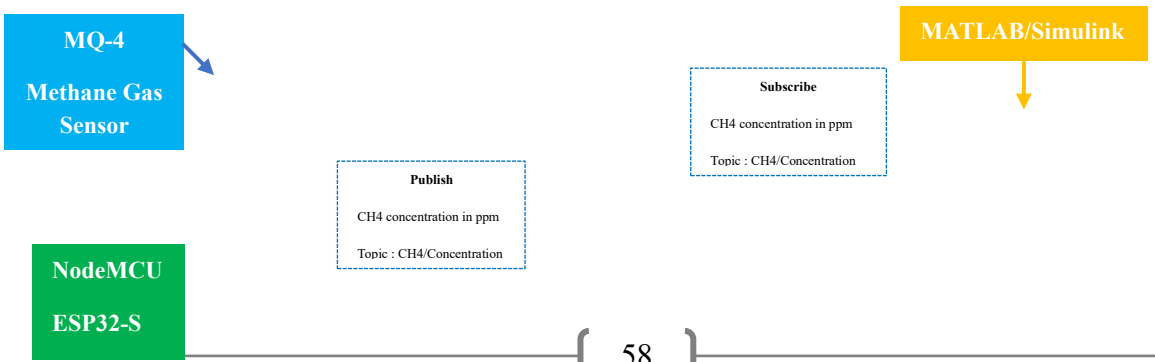


III.4.5 MQTT Protocol used in our Smart Fuel Cell Monitoring System

Message Queuing Telemetry Transfer (MQTT) is a lightweight messaging protocol designed for Internet of Things (IoT) applications. It follows a publish/subscribe communication model, where devices publish data to specific subjects and subscribers receive the data through a mediator. In our proposed monitoring system, the NodeMCU ESP32-S publishes methane concentration measurements obtained from the MQ-4 Methane gas sensor, while the corresponding MATLAB/Simulink Digital Twin subscribes to the relevant subject to receive the data in real time. MQTT was chosen due to its low bandwidth requirements, and reliability, also for suitability for real-time communication between devices and distributed applications.

III.4.5.1 Mosquitto Broker Used

Mosquitto is an open-source MQTT broker that manages message exchange between publishers and subscribers. In the proposed system, the Mosquitto broker was installed and configured on a local computer. It acts as an intermediary between the ESP32 and the MATLAB/Simulink Digital Twin environment. The broker receives methane concentration data published by the ESP32 and forwards them to the virtual SOFC model in real time. Likewise, the estimated power generated by the Simulink model can be transmitted back to the ESP32 for LED control. By managing message exchange through a publish/subscribe mechanism, Mosquitto ensures reliable communication between the physical sensing system and the Digital Twin Prototype. This what is illustrates in the figure III.15 a typical MQTT architecture used for IoT connectivity, where sensor data from the MQ-4 sensor is transmitted via Mosquitto MQTT broker to a MATLAB/Simulink DT, while the estimated SOFC power is sent back to adjust the brightness of the LED.



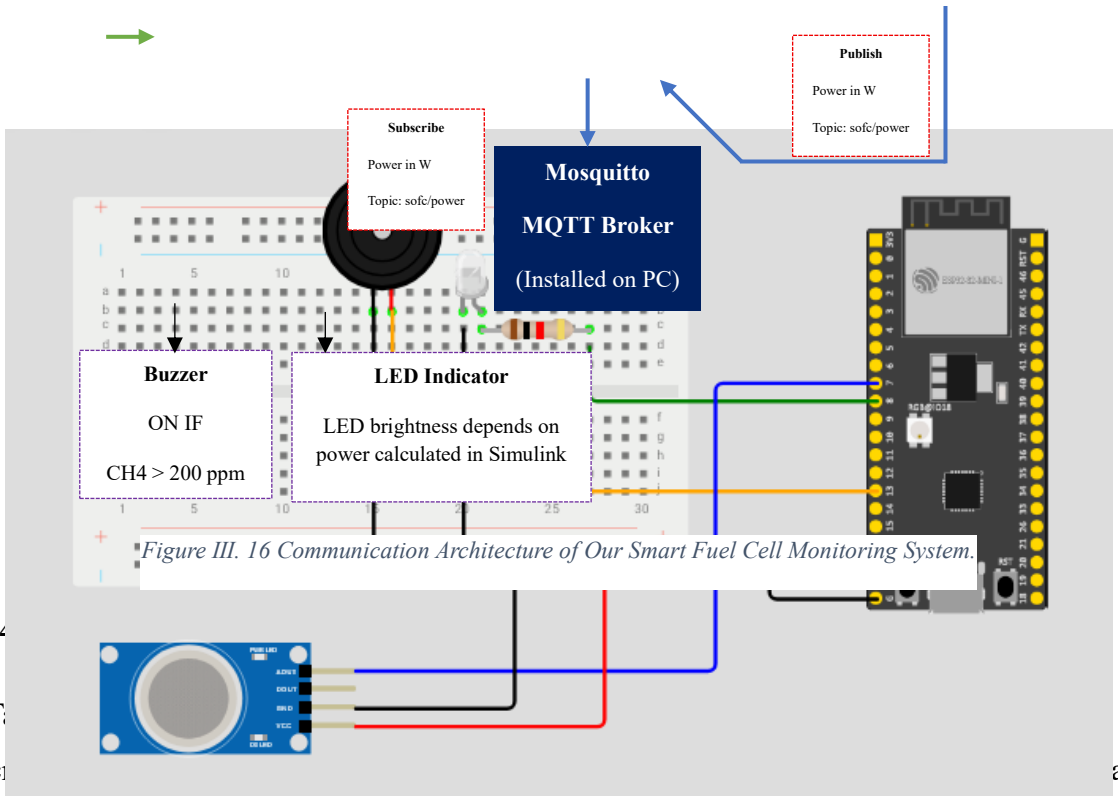


Figure III. 16 Communication Architecture of Our Smart Fuel Cell Monitoring System.

III.4

T the mic able III.9 that we followed during the assembly on the breadboard. Also, the complete wiring diagram of all the components connections are illustrated in the figure III.17 and the real realization that show in the figure III.18.

Table III. 9 Wiring connections and pin functions for our Smart Fuel Cell realization.

Component	Component pin	ESP-32S pin
MQ-4 Sensor	VCC	Vin: 5V
	GND	GND
	AO	GPIO 32
	DO	Not connected
Buzzer	(+)	GPIO 13
	(-)	GND
LED	Anode	GPIO 25 via 220 Ω
	Cathode	GND
Power	USB cable	Micro-USB port

Figure III. 17 Wiring Diagram for Our Smart Fuel Cell using The WOKWI Simulator.

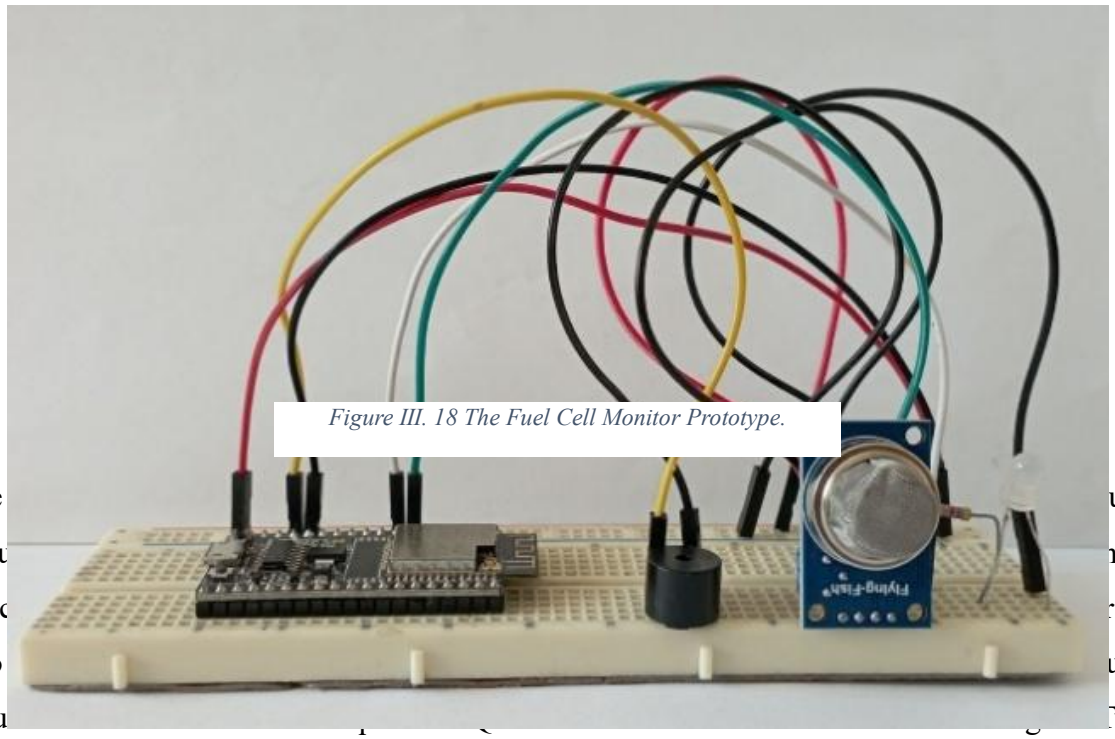
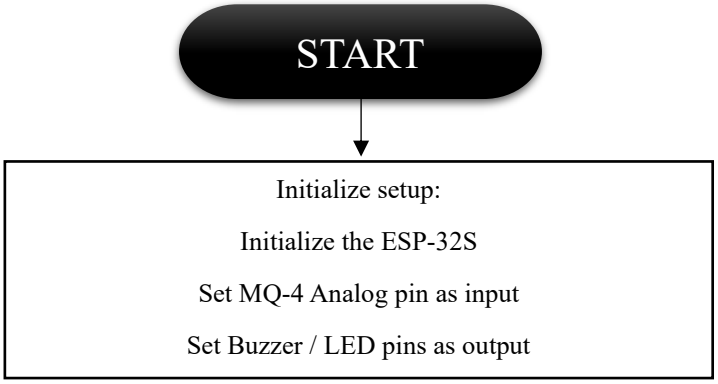


Figure III. 18 The Fuel Cell Monitor Prototype.

The usly acqu hane conc rts it into ured valu Twin subscribes to these data to estimate the electrical power generated by the virtual SOFC model. Subsequently, the calculated power value is published to the MQTT broker, where it is received by the ESP32 to adjust the LED brightness accordingly. In addition, the ESP32 continuously monitors the methane concentration and activates the buzzer whenever the methane concentration increases, providing an audible warning to the user. This bidirectional

communication ensures real-time synchronization between the physical monitoring system and the virtual SOFC Digital Twin. Figure III.19 presents the communication logic implemented in the ESP32 program.



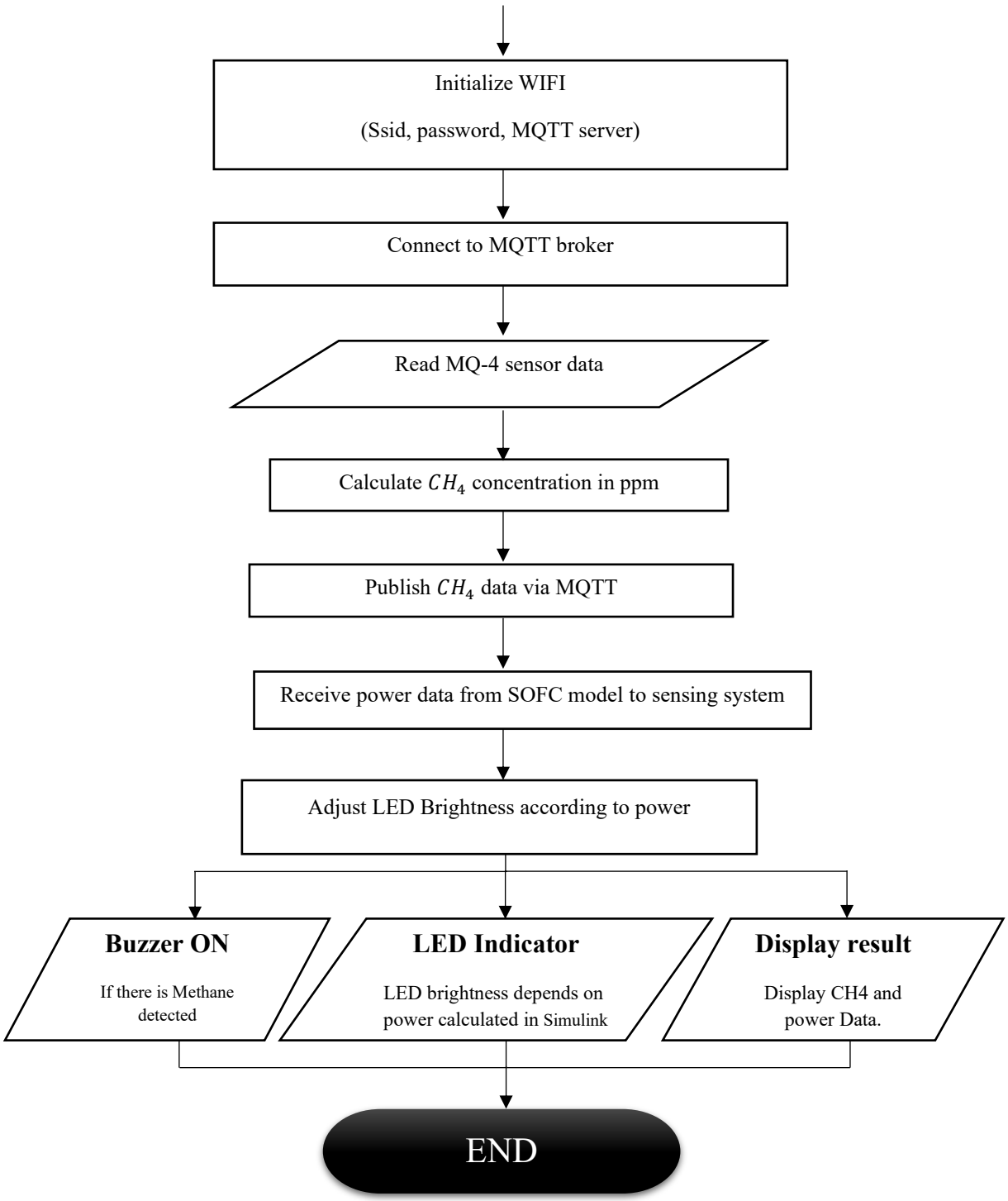


Figure III. 19 Smart Fuel Cell Monitor Code Flowchart.

III.5 Conclusion

In this chapter, we presented all the hardware and software components used in our proposed Smart Fuel Cell Monitor System. The ESP-32S acts as the brain of the system by reading the analog voltage of methane from the MQ-4 sensor through its analog input, acquiring and

transmitting the data to the Digital Twin Prototype (DTP) using the MQTT protocol communication, and controlling the buzzer and LED.

In the next chapter, we will focus on the modeling and implementation of the SOFC-DTP developed in MATLAB/Simulink for real-time monitoring and analysis of the SOFC system behavior.

CHAPTER IV:

Digital Twin of SOFC

IV.1 Introduction

The literature on digital twins distinguishes three categories of DT: Digital Twin Instance (DTI), the Digital Twin Prototype (DTP), and the Performance Digital Twin (PDT) [34]. To prove concept and ensure that systems operate flexibly and efficiently before the real-world implementation, we adopted a DTP approach. Because this allowed engineers to develop, and test the actual product in a virtual environment without needing to connect the virtual components with physical components [22].

In this chapter, we will first explain how we modeled and implemented the DT of a solid oxide fuel cell using MATLAB/Simulink. Digital twin technology was selected to simulate and better understand the fuel cell behavior under variable methane concentration. This approach enables real time monitoring, data exchange, and the interaction between our Smart Fuel Cell Monitor and the virtual model while allowing system performance to be evaluated before deployment in a real SOFC. Finally, how our model works, and the simulation results are presented and discussed to analyze the impact of methane concentration variations on the electrical and thermal behavior of the proposed system.

IV.2 Digital Twin Prototype of SOFC in MATLAB/Simulink

The Digital Twin Prototype (DT) of the Solid Oxide Fuel Cell (SOFC) developed in MATLAB/Simulink. The model is organized into several interconnected subsystems that simulate the main physical and electrochemical processes occurring inside the fuel cell. The methane concentration data received from the IoT monitoring system that we describe in chapter III are first processed and converted into fuel related inputs for the SOFC model. The model then computes key operating parameters such as current, gas dynamics gas, cell voltage, losses of the fuel cell, temperature, and electrical power output. Finally, the simulation results are displayed and transmitted to the monitoring system, enabling real-time interaction between the physical sensing environment and the virtual SOFC model. This Digital Twin Prototype provides a dynamic representation of SOFC behavior under varying methane concentrations and serves as the core component for power estimation in the proposed monitoring system. The figure IV.1 presents the architecture of the SOFC digital twin prototype developed in MATLAB/Simulink.

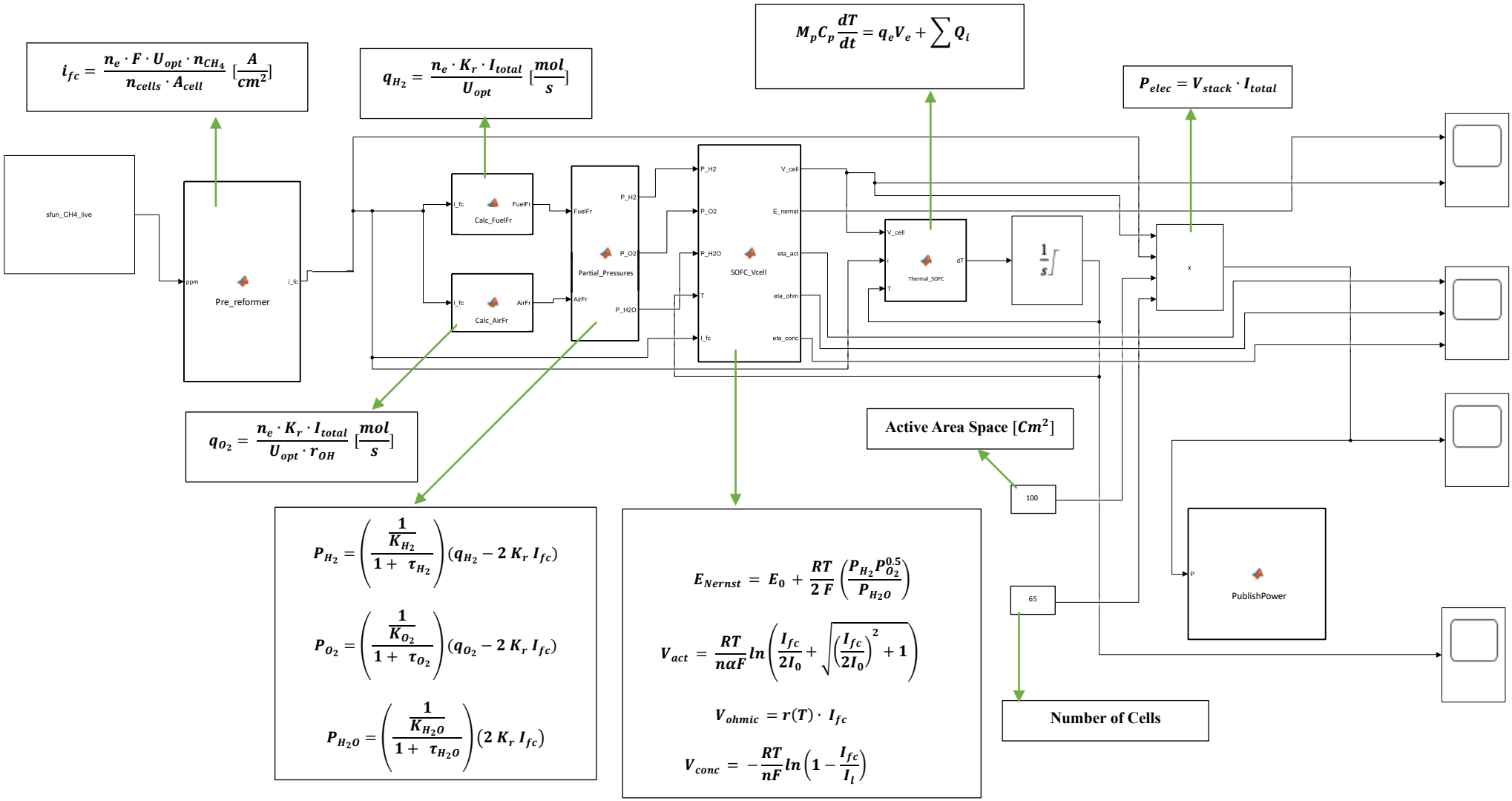


Figure IV. 1 SOFC Model in MATLAB/Simulink.

Our proposed SOFC Digital Twin is composed of many interconnected functional blocks. The process starts with the Sfun_CH4 Block, which reads the methane concentration signal coming from the datastore code, which we will explain it in the section of general system logic, this signal used as main input of the system. The Output of the first block is processed by the Pre_reformer Block, whose function is to convert the methane concentration in [ppm] into current density i_{fc} in [A/Cm^2], which used as suitable input for the fuel cell model, and the output are forwarded to the Calc_FuelFr Block, which calculates the hydrogen flow rate (q_{H_2}), and to the Calc_AirFr Block, which determines the corresponding air flow rate (q_{O_2}), these flow rates serve as inputs to the Partial_Pressures Block, where the partial pressures of hydrogen, oxygen, and water ($P_{H_2}, P_{O_2}, P_{H_2O}$) are calculated. The computed partial pressures are subsequently supplied to the SOFC_Vcell Block, which is responsible for modeling the electrochemical behavior of the fuel cell and generating outputs the Nernst voltage (E_{nernst}), Fuel Cell Voltage (V_{cell}), Activation, Concentration, Ohmic Losses. The thermal effects associated with fuel cell operation are evaluated by the Thermal_SOFC Block, which receives the operating conditions and electrical losses as inputs and produces the temperature dynamics as output. Finally, the generated power signal is transmitted through thePublishPower Block for communication, while the different outputs of the model are displayed in visualization scopes to enable real-time observation of the SOFC parameters.

IV.3 Enhanced Dynamic Modeling Approach for SOFC Systems

The mathematical heart of the Digital Twin is the SOFC model is the dynamic model proposed by Padullés 2000 [18], in the version implemented in MATLAB/Simulink by Lakshmi et al [2] and Sedghisigarchi [66]. Both papers are well known in the SOFC community and they give a complete set of equations and numerical parameters that can be re-implemented from scratch.

IV.3.1 Elements Adopted from the Reference SOFC Dynamic Model

From Padullés 2000 [18] paper we took the dynamic part of the model, the first-order time constants on the partial pressures of $H_2, O_{2,}, H_2O$, and from [67] the way the flow rates are tied to the cell current through Faraday's law. From the Lakshmi and Sedghisigarchi paper's [2, 66] we took the explicit Simulink architecture, the three irreversibility losses (activation in arcsinh form, ohmic with the Arrhenius temperature law, concentration with the limiting current). Most of the physical parameters of the model come straight from [2, 18, 66], unchanged. The parameters that describe the cell chemistry and geometry given in the table IV.1.

Table IV. 1 Parameters of the SOFC system used in the model [2, 18, 66].

Variable	Signification	Value	Unit
E_0	Standard potential	1.1	V
R	Universal gas constant	8.314	J/ (mol· K)
F	Faraday constant	96485	C/mol
U_{opt}	Optimum fuel utilization	0.85	-
r_{OH}	Ratio of hydrogen to oxygen	1.145	-
K_{H_2}	Molar valve constant for hydrogen	$8.43e^{-}4$	$K \cdot mol/(s \cdot atm)$
K_{O_2}	Molar valve constant for oxygen	$2.81e^{-}4$	$K \cdot mol/(s \cdot atm)$
K_{H_2O}	Molar valve constant for water	$2.52e^{-}3$	$K \cdot mol/(s \cdot atm)$
γ	Constant coefficient of fuel cell	0.2	ohm
β	Constant coefficient of fuel cell	-2870	K
T_0	Reference temperature for Arrhenius	973	K

The only parameters that we changed deliberately are the three response time constants $\tau_{H_2}, \tau_{O_2}, \tau_{H_2O}$. They control how fast the partial pressures react to changes in flow rate or current. The original Lakshmi values are too slow for a live demonstration. To make the dynamics observable on a timescale of a few seconds, we maintained the relative order $\tau_{H_2O} > \tau_{H_2} > \tau_{O_2}$. This, the qualitative dynamic behavior of our model remains consistent with the original Lakshmi formulation [2]. the table IV.2 shows the values response time constants that we used in our DTP of SOFC.

Table IV. 2 Parameters that can be changed in our model.

Response Time Constants	Lakshmi	Our value
τ_{H_2}	26.1 s	5.0 s
τ_{O_2}	2.91 s	3.9 s
τ_{H_2O}	78.3 s	8.0 s

Both of those paper's feed the cell with pure hydrogen, our input is methane concentration in ppm. So, we had added a chemical conversion step at the front of the model which we will be explained in the next section.

IV.4 Electrochemical Modeling of SOFC in MATLAB/Simulink

IV.4.1 Stack design parameters

The studies we used in our modeling of an SOFC fuel cell focus on report results this type of fuel cell at the single-cell level [18, 2], but in real industrial system needs a stack of my many cells in series. Two parameters in the model are our own design choices for our SOFC stack:

- The number of cells $n_{cells} = 65\ cells$
- The active area per cell $A_{cell} = 100\ cm^2$.

IV.4.2 Functional Blocks of the SOFC Digital Twin Prototype

Each of the interconnecting blocks that make up the SOFC Digital Twin Prototype is in charge of replicating a certain facet of fuel cell operation. The system is separated into three primary parts the electrochemical model, the thermal model, and a collection of auxiliary blocks this make our model easier to study. Together, these components enable the simulation of the electrical and thermal behavior of the SOFC, as well as data processing, visualization, and communication with the IoT monitoring system.

- **The Electrochemical Model:** Represents the core operation of the SOFC. It computes the primary electrical outputs, such as current, cell voltage, losses, and power generation, and mimics the electrochemical processes taking place inside the fuel cell. This block uses the fuel input, gas partial pressures, and operating conditions to estimate the electrical performance of the SOFC.
- **The Thermal Model:** Simulates the temperature dynamics of the fuel cell. It calculates the heat generated during the electrochemical reactions and the heat exchanged with the surrounding environment. This block updates the operating temperature of the SOFC, which directly influences its efficiency and electrical performance.
- **Auxiliary Blocks:** Perform supporting tasks required for the operation of the Digital Twin Prototype. These blocks handle signal conditioning, methane-to-fuel conversion calculations, parameter initialization, data processing, mathematical operations, integration functions, and the display of simulation results. They ensure proper data flow and interaction between the electrochemical and thermal subsystems.

IV.4.3 Organization of the Functional Blocks of the SOFC Digital Twin Prototype

Our SOFC Digital Twin Prototype functional blocks are arranged in three main layers. the input layer that gets data on methane concentration in real time from the MQ-4 gas sensor. The physics layer, which houses the SOFC model in charge of simulating the electrochemical and thermal behavior of our SOFC, then processes these data. Lastly, by sending the predicted power output to the monitoring system and showing the simulation findings, the output layer offers feedback and visualization features. The table IV.3 illustrates the block name, type, and the layer of each block we used to build our DTP SOFC.

Table IV. 3 The block names, types, and layers used in our SOFC model.

	Block name	Type	Function
Input	sfun_CH4_live	Level-2 MATLAB S-Function	Auxiliary block
	Pre-reformer	MATLAB Function	Electrochemical model
Physics	calc_FuelFr	MATLAB Function	Electrochemical model
	calc_AirFr	MATLAB Function	Electrochemical model
	Partial_pressures	MATLAB Function	Electrochemical model
	SOFC_Vcell	MATLAB Function	Electrochemical model
	Thermal_SOFC	MATLAB Function	Thermal model
	Integrator	Continuous-time integrator	Auxiliary block
	Stack_Puissance	Product	Auxiliary block
Output	Scopes	Standard scope	Auxiliary block
	Publish_Power	MATLAB Function	Auxiliary block

IV.4.3.1 Input layer

The input layer forms an interface between the physical sensing system and the digital twin prototype. It receives real-time methane concentration measurements coming from the MQ-4 gas sensor via an IoT communication system and provides the necessary input data for the solid oxidized fuel cell prototype.

IV.4.3.1.1 Level-2 MATLAB S-Function Block

This is a Level-2 MATLAB S-Function (**Sfun CH_4 live).** Its only job is to read the latest filtered ppm value from the shared datastore and put it on its output port at every Simulink step. It has no input port - the data comes from outside the model, via the MQTT callback. We will explain this part in the chapter V.

IV.4.3.1.2 Pre-reformer Block

As we mentioned in the first chapter when we discussed how solid oxide fuel cells works when the input is methane and not pure hydrogen, before entering into the basic transformation that take place at the anode and cathode level. The role of Pre-reformer block here is that it converts the methane concentration measured by the MQ-4 sensor into an equivalent fuel cell current density by representing the methane to hydrogen reforming process and applying Faraday's law.

This block is the chemical interface between the methane sensor and the rest of the model. It implements the SR (I.4) & WGR (I.6), the overall reaction that we explained in the first chapter [16]:

$$\boldsymbol{CH_4 + 2\,H_2O \rightarrow CO_2 + 4\,H_2} \quad \textit{Eq I.7}$$

At the anode:

$$\boldsymbol{4\,H_2 + 4\,O^{2-} \rightarrow 4\,H_2O + 8\,e^-} \quad \textit{Eq I. 8}$$

At the cathode:

$$\boldsymbol{2\,O_2 + 8\,e^- \rightarrow 4\,O^{2-}} \quad \textit{Eq I. 9}$$

The input of the pre-reformer block is the methane concentration in ppm (parts per million) received from Block 1, while the output is the fuel cell current density i_{fc} [A/cm²], but before we compute the current density from the concentration of methane, we must convert from ppm to methane molar flow rate in [mol/s] using the ideal gas law. First, the total molar flow rate ($\dot{n}_{tot}$) is computed from the inlet volumetric flow rate ($\dot{V}$), pressure (P), and temperature conditions (T). The methane concentration is then converted into a molar fraction by dividing the concentration in ppm by (10^6) ,and multiplied by the total molar flow rate ($\dot{n}_{tot}$) to obtain the methane molar flow rate (n_{CH_4}). This flow rate is subsequently used to determine the current density supplied to the SOFC model.

In our ideal case we assume that one mole of CH_4 produces four moles of H_2. This assumption in the Pre_reformer block. From it, the cell current comes out of Faraday's law [67] applied with 8 electrons per molecule (4 H_2 times 2 electrons each), the equation IV.1 shows all.

$$\boldsymbol{i_{fc} = \frac{n_e * F * U_{opt} * n_{CH_4}}{n_{cells} * A_{cell}}} \quad \textit{Eq IV.1}$$

The logic we explained in how the pre-reformer block works is illustrated in the figure IV.2 showing the logic used to program it.

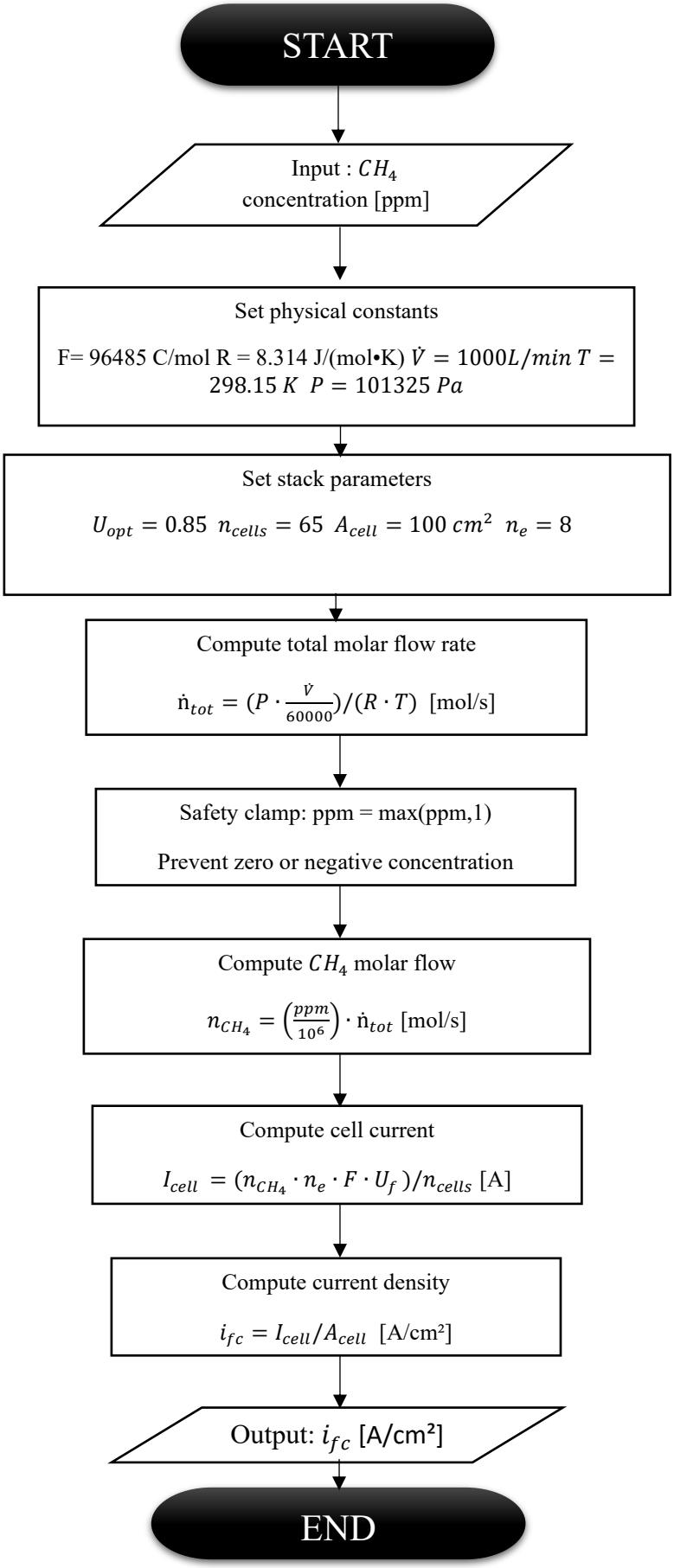


Figure IV. 2 Pre-reformer Block Flowchart Code.

The table IV.4 includes all parameters used for the calculation in pre-reformer block.

Table IV.4 The Parameters used in the Pre-reformer Block.

Parameter	Value	Unit	Description
P	101325	Pa	Absolute Pressure
$\dot{V}$	1000	L/min	Volumetric flow rate
R	8.314	J/ (mol· K)	Ideal Gas Constante
T	273.15	K	Absolute Temperature
n_e	8	-	Number of electrons.
F	96485	C/mol	Faraday constant.
U_{opt}	85	-	Optimum fuel utilization [2].
n_{cells}	65	Cells	Number of cells in series.
A_{cell}	100	cm²	Active area per cell.

IV.4.3.1.3 Block of Calc-FuelFr

This bloc works on computes the molar flow rate of H_2 fed to the anode, from the cell current, using the simplified steady state formula in according with the equation cited in [68] according to Padullés 2000, that given with the equation IV.2.

$$q_{H_2} = \left(\frac{2\,K_r\,I_{total}}{U_{opt}}\right) \qquad \textit{Eq IV.2}$$

The reaction constant K_r is the bridge between the electrical world in [A] and the chemical world (molar flow in [mol/s], and it used to calculate the partial pressures of gases (H_2, O_2), and water [18]. The factor $2 * K_r$ says that two H_2 are consumed per O_2, and division by U_{opt} accounts for the fact that only a fraction U_f of the fed H_2 is electrochemically used (the rest leaves with the anode exhaust) [68]. The figure IV.3 illustrates the logic that we have explained in this block.

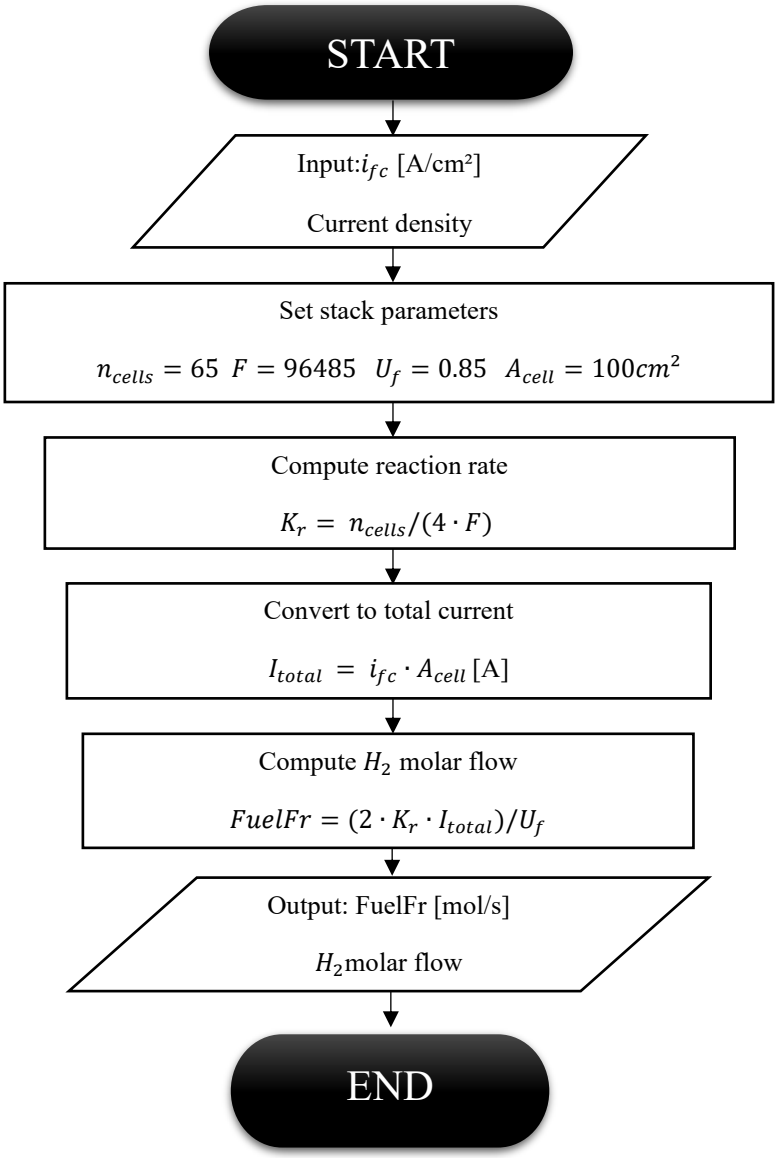


Figure IV. 4 Calc-FuelFr Block Flowchart Code.

IV.4.3.1.4 Block of calc-AirFr

The oxygen flow is linked to the hydrogen flow through the ratio r_{OH} = 1.145, which comes from [18, 2, 68] and corresponds to a slight excess of air on the cathode for cooling and to prevent local oxygen starvation, we used this following equation:

$$q_{O_2} = \frac{q_{H_2}}{r_{OH}} = \frac{2K_r I_{total}}{U_{opt} r_{OH}} \qquad \textbf{Eq } IV.3$$

The figure IV.4 illustrates the same logic we used in the pervious block but this time to calculate the hydrogen flow using the ratio.

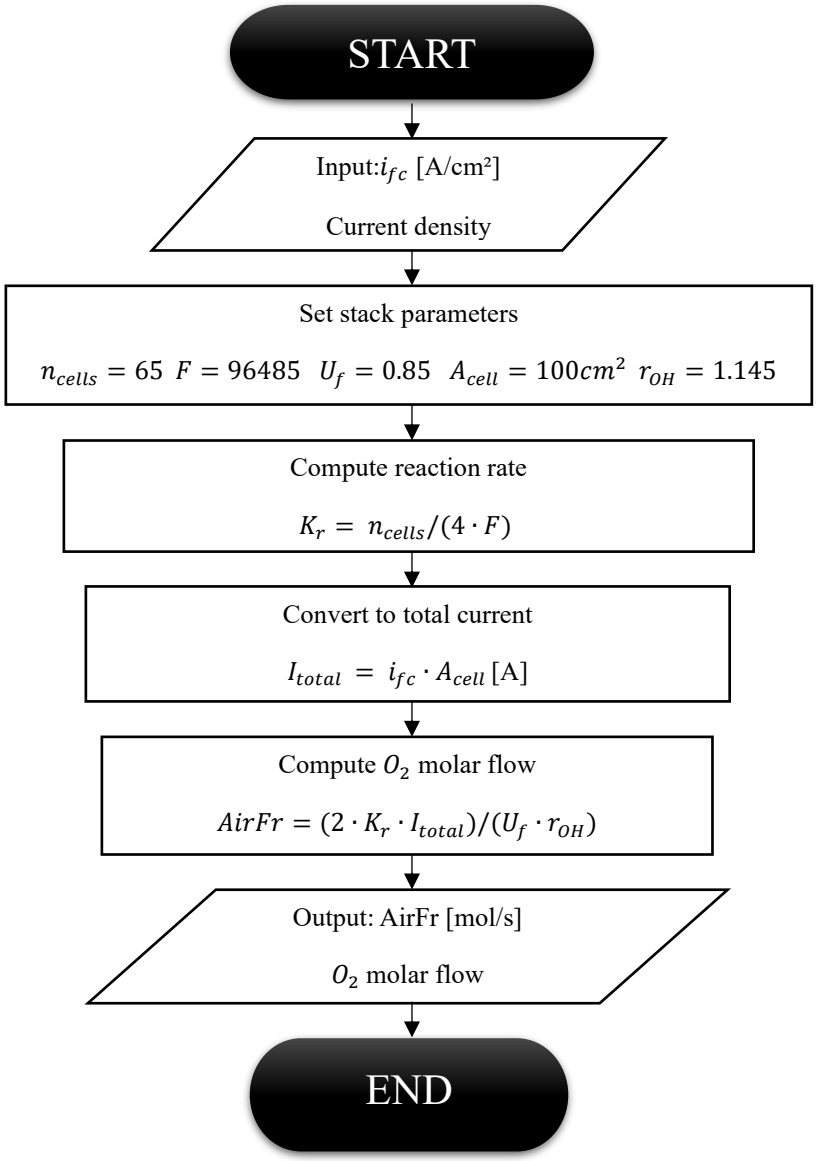


Figure IV. 5 Calc-AirFr Block Flowchart Code.

IV.4.3.1.5 Partial Pressures block

This is the dynamic heart of the whole model. It implements equations 3, 4 and 5 of [2], which describe how the partial pressures $P_{H_2}, P_{O_2}, P_{H_2O}$ evolve in time when the flow rates and current change. Each pressure follows a first-order ordinary differential equation with its own response time in seconds as we mentioned in the first chapter. These partial pressures are important and are used to calculate the Nernst potential (E_{nernst}) and fuel cell voltage in the following block [2]. We used the equations that we mentioned in chapter I in how to calculate the partial pressures.

MATLAB Function blocks do not support Laplace transfer functions, that why discretized the differential equation with the explicit Euler method because it's generally preferred for better

numerical stability, and store the state in three persistent variables [69]. The persistent variables keep their value between simulation steps, which is how the integration accumulates over time.

The first step was to convert the equations, originally expressed as Laplace-domain transfer functions, into the standard form of a first-order dynamic system given by the equation

$$\tau \frac{dP(t)}{dt} + P(t) = P_{target} \qquad \textit{Eq IV. 4}$$

Then we used Euler's method to convert it into a digital format, using the equation IV.5:

$$P_{K+1} = P_k + dt \cdot \frac{P_{target} - P_k}{\tau} \qquad \textit{Eq IV. 5}$$

The figure IV.5 illustrates the logic we used to program the Partial Pressures Block.

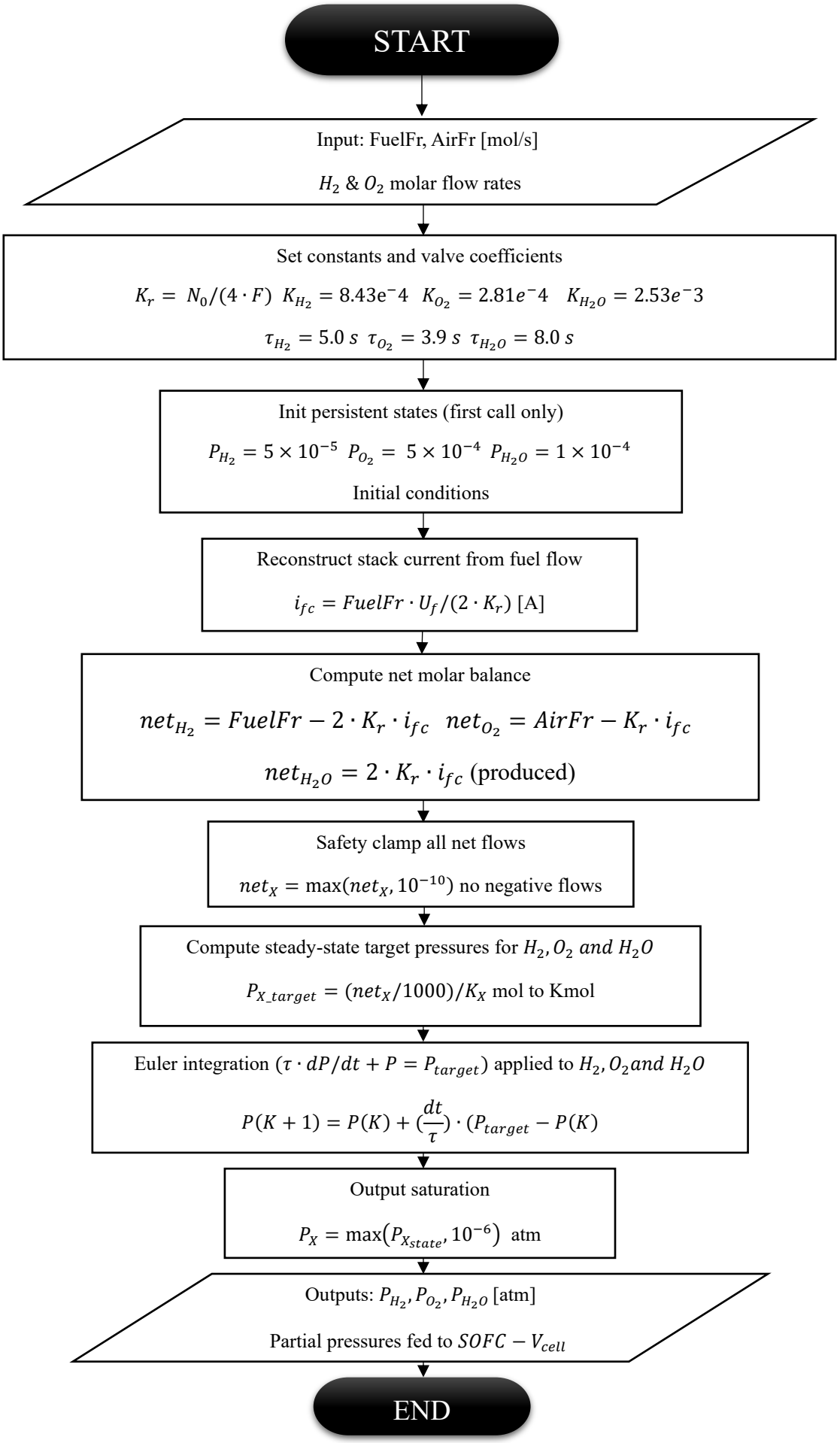


Figure IV. 6 Partial Pressures Block Flowchart Code.

IV.4.3.1.6 SOFC Voltage cell Block (SOFC Vcell)

This block computes the actual cell voltage from the partial pressures and the current. It implements Lakshmi equation 1 [2] given by the equation IV.6:

$$\boldsymbol{V_{fc} = E_{nernst} - V_{act} - V_{ohmic} - V_{conc}} \qquad \textbf{Eq } \textit{IV. 6}$$

With [2]:

- The Nernst voltage E_{nernst} it's depending on the partial pressures.
- The activation loss V_{act} in Bulter-Volmer Arcsinh form.
- The ohmic loss with the Arrhenius temperature law.
- The concentration loss V_{conc} with the limiting current.

To proof the concept in the initial phase of our model we determine the value of the exchange current density $I_0 = 0.25e\text{-}3\ A/cm^2$, and the limiting current density $I_l = 1.5\ A/cm^2$.

In this block we have five outputs $V_{cell}, E_{nernst}, V_{act}, V_{ohm}, and\ V_{conc}$ are wired separately so we can put a scope on each loss to see their behavior in real-time simulation, and the equations that we used are mentioned in chapter I. The logic that we used for modeling this block is illustrated in figure IV.6.

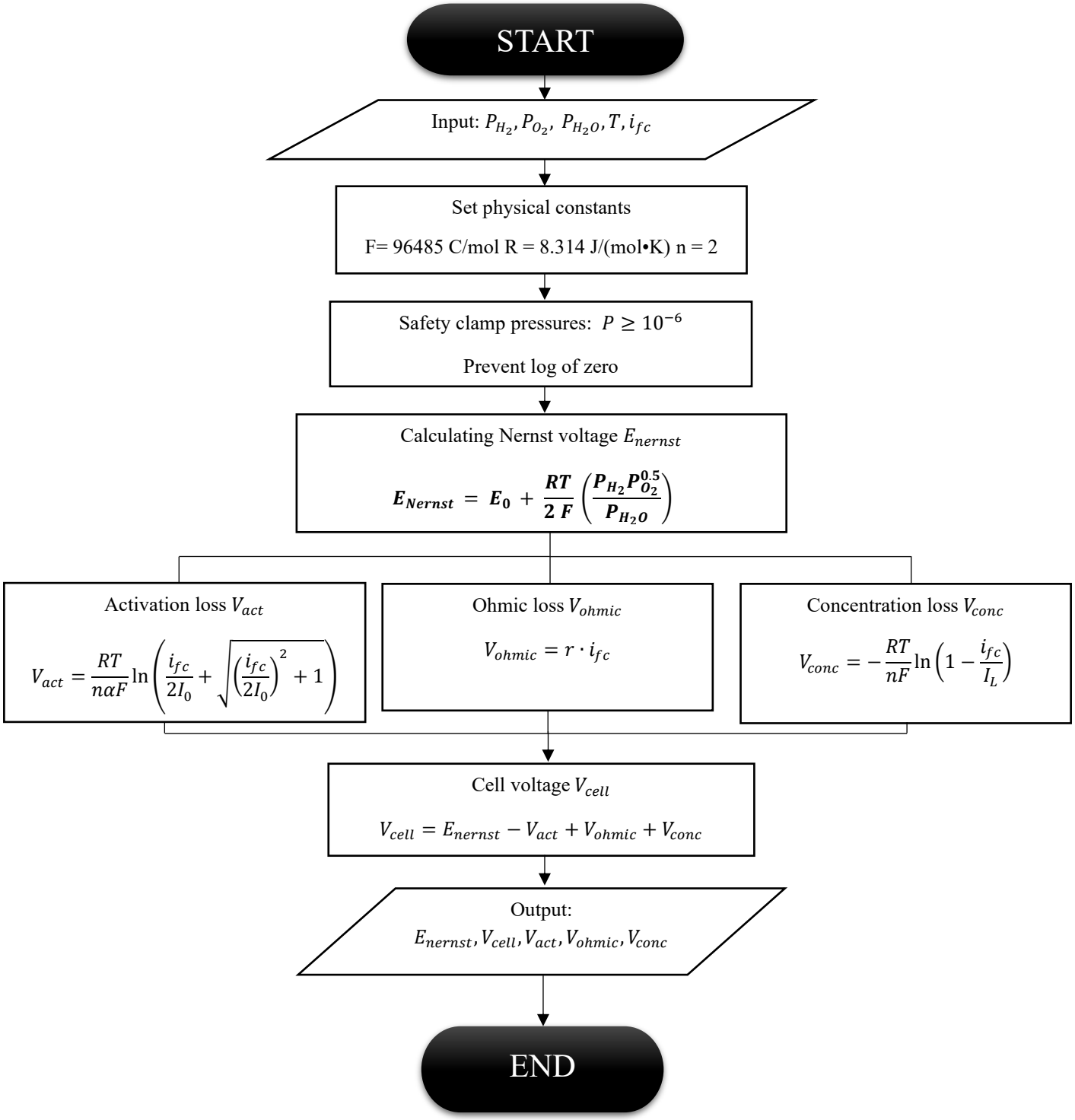


Figure IV. 8 SOFC Vcell Block Flowchart Code.

IV.4.3.1.7 Thermal SOFC block

The main idea of the Thermal SOFC block is that any chemical energy not converted to electricity becomes heat.

The thermal modeling of the SOFC starts from a fairly simple observation: the fuel cell is not a cold box it heats up, and this heat changes everything. Sedghisigarchi and Feliachi [66] establish the energy balance of each cell using the formula given by the equation IV.7:

$$\boldsymbol{M_p C_p \frac{dT}{dt} = q_e V_e + \sum Q_i} \qquad \textit{Eq IV. 7}$$

The table IV.5 illustrates the parameters of the Energy Balance Equation and significations.

Table IV.5 Parameters and their significations [66].

symbol	Signification
M_p	Mass of the cell unit (in kilograms).
V_e	Volume of the cell unit (m^3).
C_p	Heat capacity of the cell unit (in Joules per kilo-gram. Kelvin).
T	Temperature of the cell unit (in Kelvin).
q_e	Heat generated from the electrochemical reaction per volume unit.
Q_i	Total heat (J) which is the summation of the q_s and other conductive, convective, and radiative heat between cell unit and separators.

This equation simply states that the change in energy stored in the cell its energy produced by the electrochemical reaction + heat exchange with the outside [66].

Identifying $q_e V_e$: heat produced by the reaction

The overall reaction inside the SOFC is given by the equation IV.8:

$$\boldsymbol{H_2 + ½\, O_2 \rightarrow H_2O} \qquad \textit{Eq IV. 8}$$

The total chemical energy released per mole of H_2 is described by the reaction enthalpy ΔH. The thermoneutral voltage ($V_{th} or\ E^H$), which is determined by the equation IV.9, is equivalent to this energy at the operating temperature of about 800°C [15].

$$\boldsymbol{E^H = V_{th} = \frac{-\Delta H}{nF} \quad \Rightarrow \quad \Delta H = -nFV_{th}} \qquad \textit{Eq IV. 9}$$

Where $n = 2$ is the number of electrons exchanged per H_2, and F is the Faraday's constant. The value of $V_{th} = 1.48\ V$ at high temperature [15].

For a stack of N_0 cells in our case we chose the number of cells = 65, carrying a total current I_{total}, the total chemical power released is given by the equation IV.10 [15]:

$$\boldsymbol{P_{chemical} = I_{total} \cdot N_0 \cdot V_{th}} \qquad \textit{Eq IV. 10}$$

However, part of this energy is converted into electricity and leaves the stack, which is illustrates in the equation IV.11 [15]:

$$\boldsymbol{P_{elec} = I_{total} \cdot N_0 \cdot V_{cell}} \qquad \textit{Eq IV. 11}$$

By the conservation of energy, the heat actually dissipated inside the cell is the difference between the chemical and electrical power, the equation IV.12:

$$\boldsymbol{Q_{heat} = P_{chemical} - P_{elec} = I_{total} \cdot N_0 \cdot (V_{th} - V_{cell})} \quad \textit{Eq IV. 12}$$

Identifying $\sum Q_i$: thermal losses to the surroundings

In [66] authors state that Qi represents heat exchanges by conduction, convection and radiation between the cell and its separators/environment. For a simplified model (stack inside a furnace at temperature T_{env}. Newton's law of cooling gives:

$$\boldsymbol{Q_{conv} = h(T - T_{env})} \qquad \textit{Eq IV. 13}$$

Where the h is the heat transfer coefficient [W/K], T is the cell temperature (state variable), and T_{env} *the ambient or environment temperature* [K]. This term is negative in the balance (heat leaving the cell), the equations IV.14 and IV.15 show all:

$$\boldsymbol{\sum Q_i = -Q_{conv} = -h(T - T_{env})} \qquad \textit{Eq IV. 14}$$

$$\boldsymbol{M_p C_p \frac{dT}{dt} = Q_{heat} - Q_{conv}} \qquad \textit{Eq IV. 15}$$

This is exactly what the Thermal-SOFC block computes. Its output dT is then integrated over time by the Simulink Integrator block to obtain T(t), with an initial condition of 1073 K and physical saturation limits between 873 K and 1273 K. The figure IV.7 show the logic that we used in this block to code it.

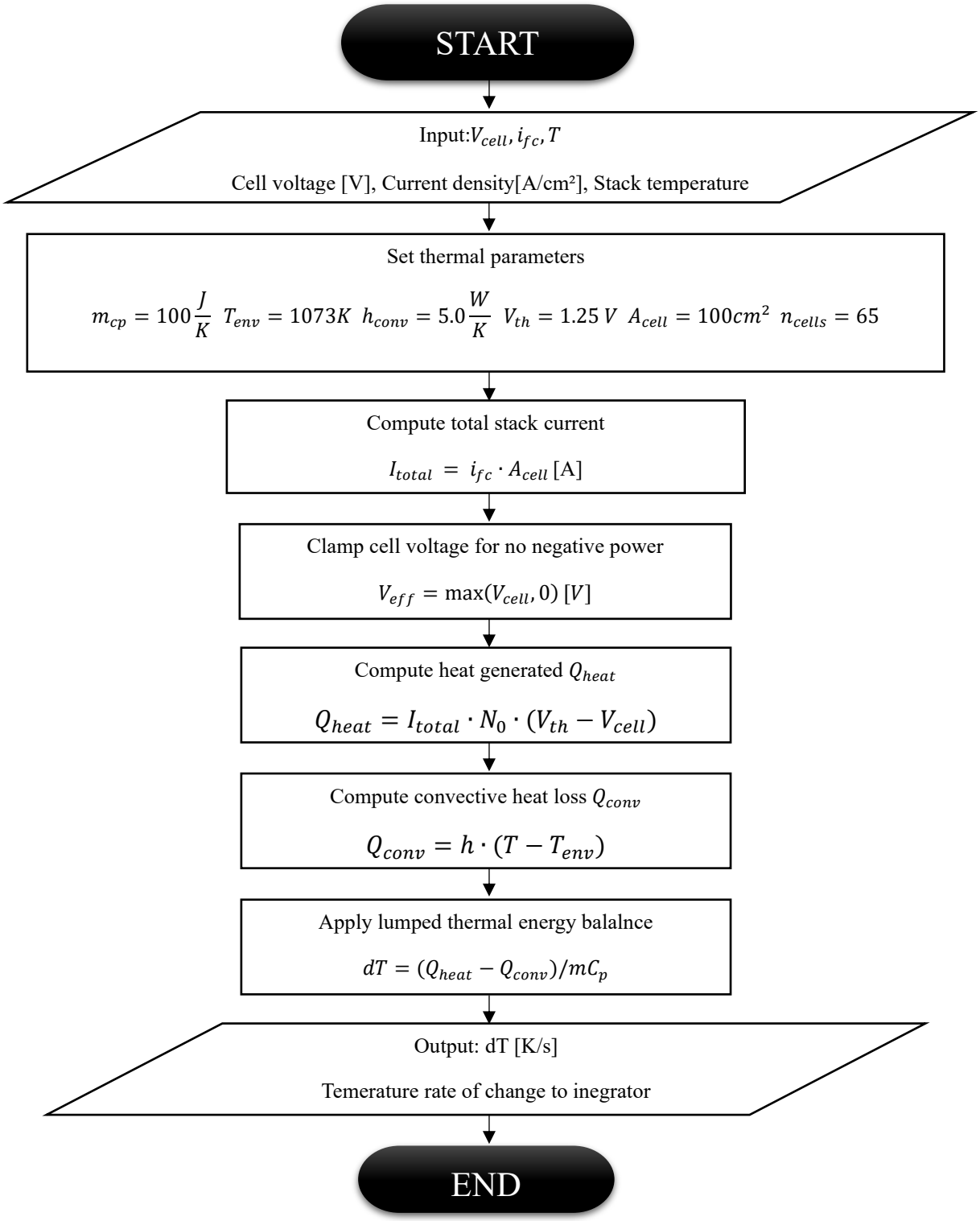


Figure IV. 9 Thermal SOFC Block Flowchart Code.

IV.4.3.1.8 Total stack power block

This block helps us to get the voltage electrical power in watts [W], by multiplying the cell voltage V_{cell} by the number and the surface area of the cells and finally the density current i_{fc} [A/cm²] [70], according to the equation IV.16

$$\boldsymbol{P_{elec} = V_{stack} \cdot I_{total} = (n_{cells} \cdot V_{cell}) \cdot (i_{fc} \cdot A_{cell})} \qquad \textbf{Eq IV. 16}$$

IV.4.3.1.9 Publish power block

This is the block that we added to close the loop. It is a tiny MATLAB Function block with one input (P_{elec}) and no output. It writes the latest power value into the shared datastore by calling aftafuel_datastore ('write_power', P) this power value controls the brightness of the LED in our Smart Fuel Cell Monitoring System, we will discuss the logic of the datastore code in the section of general system logic. The figure IV.8 show the how we program the Publish Power Block.

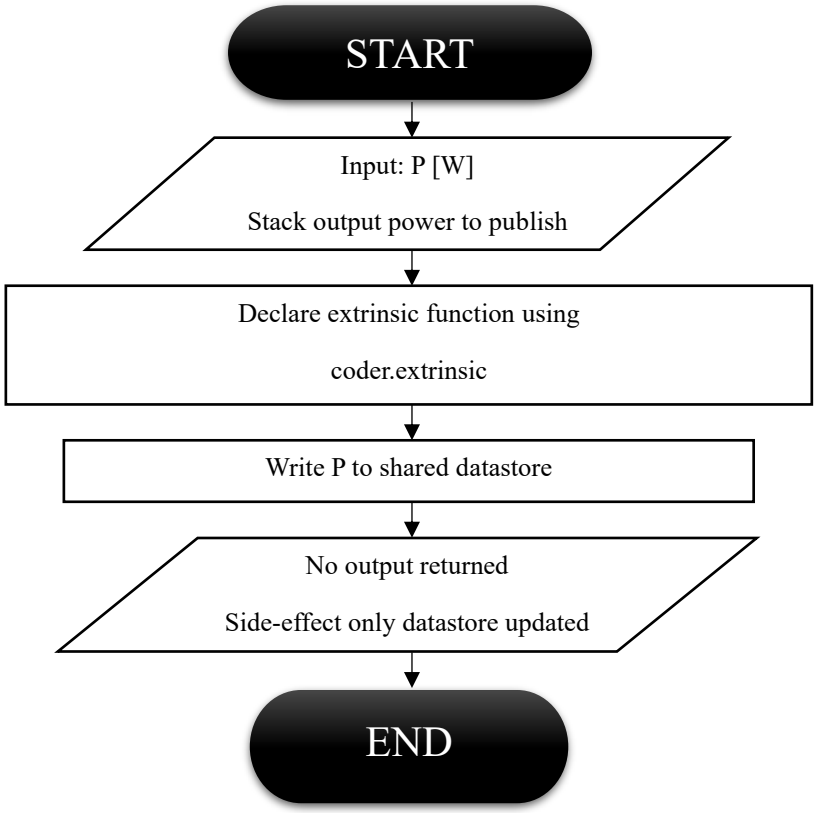


Figure IV. 11 Publish Power Block Flowchart Code.

IV.4.3.1.10 Scopes blocks

In our model built in MATLAB, we used four scopes to visualize the behavior in the simulation in real time, we will explain the results obtained from the SOFC DT in the section of results and discussion:

- Scope 1 for: V_{cell}, E_{nernst}.
- Scope 2 for: $V_{act}, V_{ohmic}, V_{conc}$.
- Scope 3 for: Stack power.
- Scope 4 for: Temperature.

Now, we presented the function of each block of our proposed SOFC DT, which provides a virtual representation of the fuel cell behavior, and enables real-time simulation, and analysis without requiring a physical SOFC. In this next section we will present the general logic system of our project, and how it operates step by step, highlighting the flow of data from methane detection to the generation of simulation results.

IV.5 General System Logic

Our project is built on a real-time digital twin Internet of Things architecture intended for solid oxidized fuel cell (SOFC) simulation and methane monitoring. A physical methane source, an ESP32-based sensing system, MATLAB communication scripts, and a Simulink digital twin model are all connected by the overall logic. Methane data is gathered, communicated, processed, and used to simulate SOFC performance in the system's closed-loop operation. The results are then provided back for visualization.

IV.5.1 MATLAB Communication Scripts

On this section we will present the MATLAB communication scripts developed for the implementation of the proposed system. Real-time data exchange between the digital twin model in Simulink and the IoT-based monitoring system becomes largely possible thanks to these scripts. The three main MATLAB programs are:

- Shared Data Management.
- Simulink Interface Level-2 MATLAB S-Function.
- IoT Data Acquisition and Communication Layer.

IV.5.1.1 Shared Data Management:

A central datastore function is used as a shared memory between all system components. because it stores the recent methane concentration, filtered values, power output, timestamps. The entering of concentration of methane signal is smoothed and noise is decreased by using an exponential moving average filter. The MATLAB script, Simulink model, and MQTT network may communicate with each other thanks to this datastore, which also guarantees data consistency.

IV.5.1.2 Simulink Interface Level-2 MATLAB S-Function

This this the first block in our model in Simulink that we mentioned in chapter IV. This Level-2 S-function acts as an interface between MATLAB and the Simulink Digital Twin model. It continuously reads the latest methane concentration from the shared datastore and provides it as a real-time input to the SOFC model. Dynamic response to variations in methane content is made possible by this simulation. The S function only handles data transfer and synchronization between the Simulink program and the MATLAB environment; it does not carry out any electrochemical calculations.

IV.5.1.3 IoT Data Acquisition and Communication Layer

Communication between the digital environment and the physical system is handled by MATLAB's first component. This script connects to our installed Mosquitto MQTT Broker, subscribes to the ESP32 module transmission of methane concentration data, and continuously receives readings in real time. Additionally, it allows two-way communication by publishing the computed energy values to the ESP32 module. Additionally, this layer controls the communication workflow across all components and guarantees system configuration.
The real-time communication between a digital twin environment and a physical monitoring unit power the project system. To guarantee data management, transmission, and interaction with the Simulink model, these three MATLAB scripts are utilized. Data on methane is measured by the sensor device and sent over MQTT to the MATLAB environment, where it is received, stored, and processed. Real-time fuel cell behavior modeling is made possible by sending the processed data via the S-Function interface to the Simulink SOFC model. Important variables including voltage, loss, temperature, and output power are computed using the model. Ultimately, a closed-loop communication structure between the system's digital and physical components is established by sending the generated power to the ESP32 module via MQTT.

IV.5.2 Fuel Cell Digital Twin Operation

In this section, we describe the operation of our smart fuel cell monitoring system with digital twin of SOFC. Within a single framework, the system combines methane gas detection, IoT, and real-time fuel cell simulation. A solid-state fuel cell (SOFC) digital twin prototype runs real-time simulations in a MATLAB/Simulink environment after data from methane gas detection is sent over an Internet of Things communications infrastructure. The essential phases of system functioning, from data collection to simulation result display and feedback production, are depicted in the following figures.

IV.5.2.1 Methane Detection and Data acquisition

First, our Smart Fuel Cell Monitor System detect methane and convert it into an analog voltage by the MQ-4 gas sensor this signal is then acquired by the ESP-32S microcontroller for further processing through its analog to digital converter (ADC). The figure IV.9 shows the simulation of the methane using a lighter.

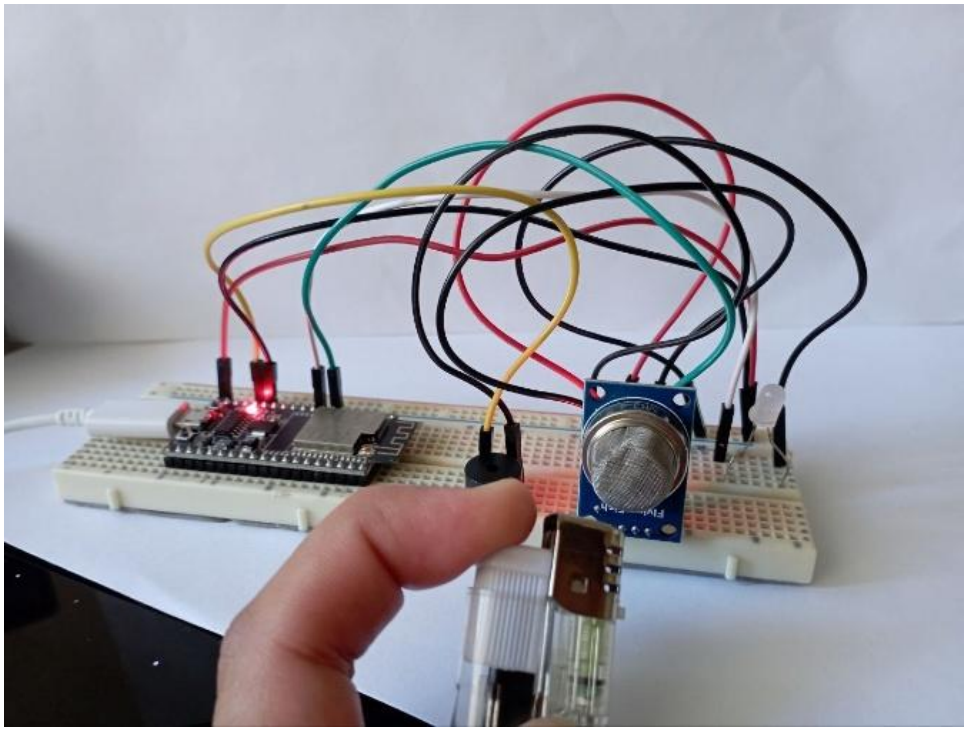


Figure IV. 9 Simulation of Methane using a Lighter.

IV.5.2.2 Data Transmission Via MQTT Broker

In this step the MQTT protocol is used to send the methane concentration data over Wi-Fi. The information transfer between the MATLAB environment and the physical system is managed by the Mosquitto broker. The figure V.10 shows the interface of the Mosquittto broker and the transmitted data using the following command in command window after running the Broker.

```
mosquitto_sub -h localhost -v -t "#"
```

```
PS C:\WINDOWS\system32> mosquitto_sub -h localhost -v -t "#"
aftafuel/ch4  "ppm":108.04
aftafuel/ch4  "ppm":107.80
aftafuel/ch4  "ppm":107.57
aftafuel/ch4  "ppm":107.87
aftafuel/ch4  "ppm":107.60
aftafuel/ch4  "ppm":107.77
aftafuel/ch4  "ppm":107.80
aftafuel/ch4  "ppm":107.80
aftafuel/ch4  "ppm":107.88
aftafuel/ch4  "ppm":107.71
aftafuel/ch4  "ppm":107.66
aftafuel/ch4  "ppm":107.64
aftafuel/ch4  "ppm":107.64
aftafuel/ch4  "ppm":107.74
aftafuel/ch4  "ppm":107.57
aftafuel/ch4  "ppm":107.46
aftafuel/ch4  "ppm":107.17
aftafuel/ch4  "ppm":106.79
aftafuel/ch4  "ppm":106.65
aftafuel/ch4  "ppm":107.12
aftafuel/ch4  "ppm":106.96
aftafuel/ch4  "ppm":107.09
aftafuel/ch4  "ppm":106.94
aftafuel/ch4  "ppm":106.91
aftafuel/ch4  "ppm":106.94
```

Figure IV. 10 The Mosquitto Broker Interface and the Transmitted Data.

IV.5.2.3 Data Transmission Via MQTT Broker

In this step MATLAB script receives the data sent by the ESP32 in to the broker and stores it in a shared memory (datastore). This step makes the measurements available for real-time simulation. The figure IV.11 illustrates the shared data in the command window of MATLAB.

```
Command Window
  AftaF Digital Twin.....MQTT work

[1/5] Datastore initialise.
[2/5] Connecting to MQTT 192.168.100.15:1883 ...
      OK - Connecte au broker MQTT
[3/5] Subscribing to "aftafuel/ch4" ...
   Listening for ESP32 data
[4/5] Client MQTT active in workspace.
[5/5] Power publication timer started  (1 Hz).

Ready to start  the simulation now
  File    : Aftafuel_SOFC.slx
  Solver     : Fixed-step odel s
[MQTT] CH4 =   131.0 ppm
[MQTT] CH4 =   138.9 ppm
[MQTT] CH4 =   144.8 ppm
[MQTT] CH4 =   148.9 ppm
[MQTT] CH4 =   150.8 ppm
[MQTT] CH4 =   152.4 ppm
[MQTT] CH4 =   150.9 ppm
[MQTT] CH4 =   150.2 ppm
```

Figure IV. 11 The transmitted Data in the Command Window of MATLAB.

IV.5.2.4 Digital Twin Simulation

The SOFC model uses the received methane concentration data to calculate the various system parameters, including Nernst voltage, electrochemical losses, fuel cell temperature, and power generated. The figure IV.12 shows the SOFC model currently running.

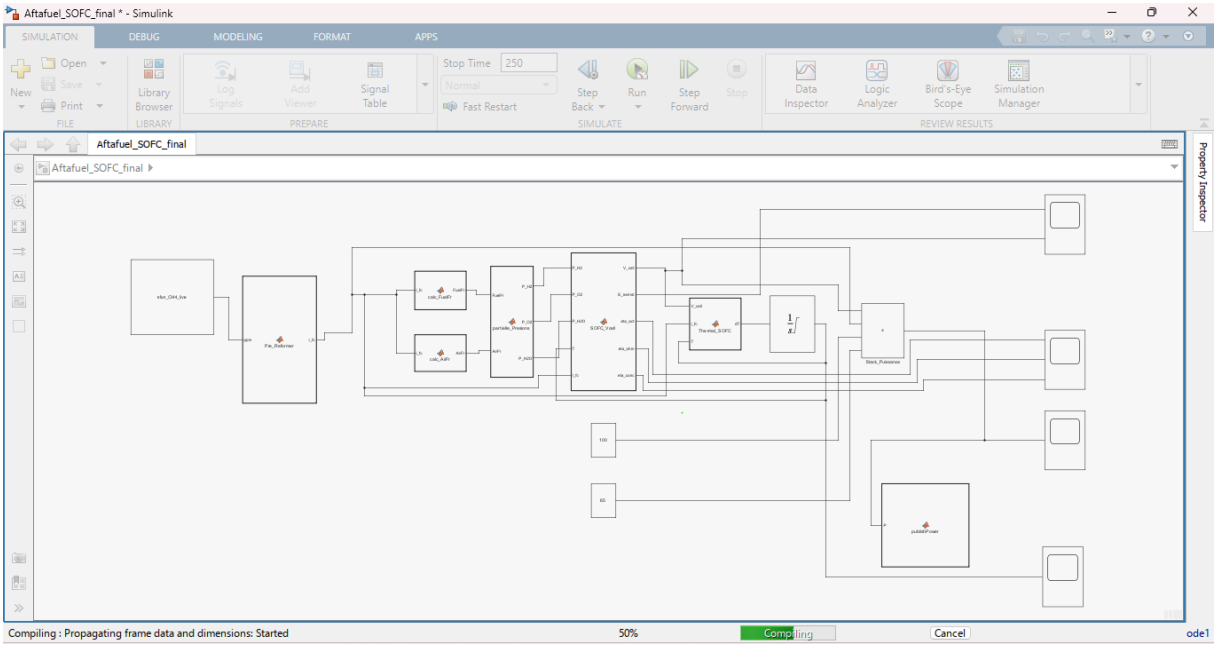


Figure IV. 12 SOFC DT Model Currently Running.

IV.5.2.5 Feedback from the SOFC Digital Twin

MQTT is used to transmit the computed power back to the ESP32. The LED brightness can be changed using this data, which also gives visual feedback on the state of the device. Figure IV.13 illustrates a high LED brightness level, which indicates a high-power value generated by the fuel cell model as a result of the increased methane concentration.

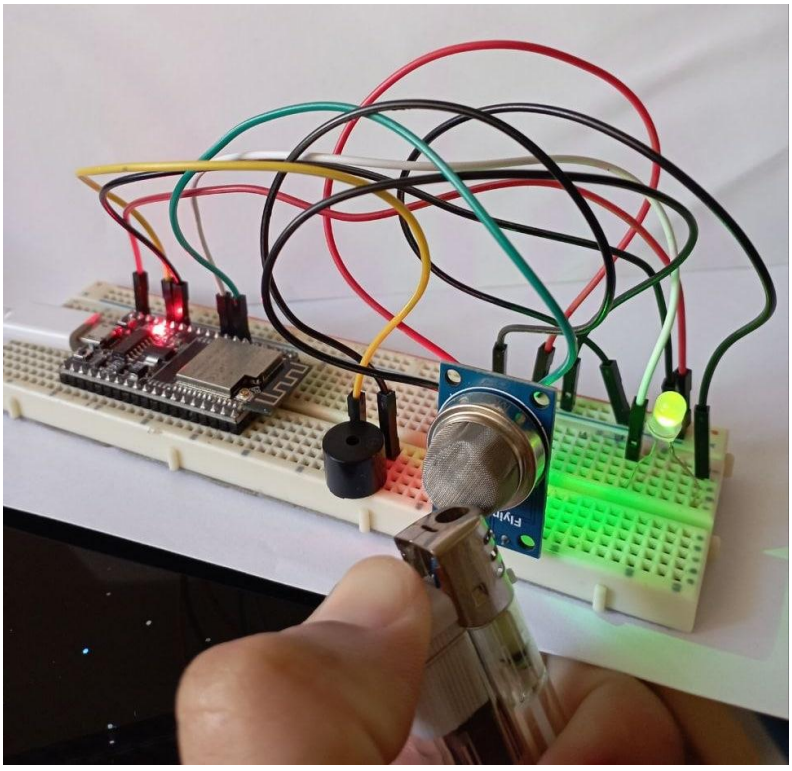


Figure IV. 13 LED at High Brightness Due to Methane Detection.

IV.5.2.6 Results and Discussion

The analysis of the dynamic behavior of our SOFC DT is made possible by the display of the simulation results of Nernst Voltage, Fuel Cell Voltage, Losses, Temperature, and Electrical Power as curves using scopes blocks. the real-time simulation is given by figure IV.14.

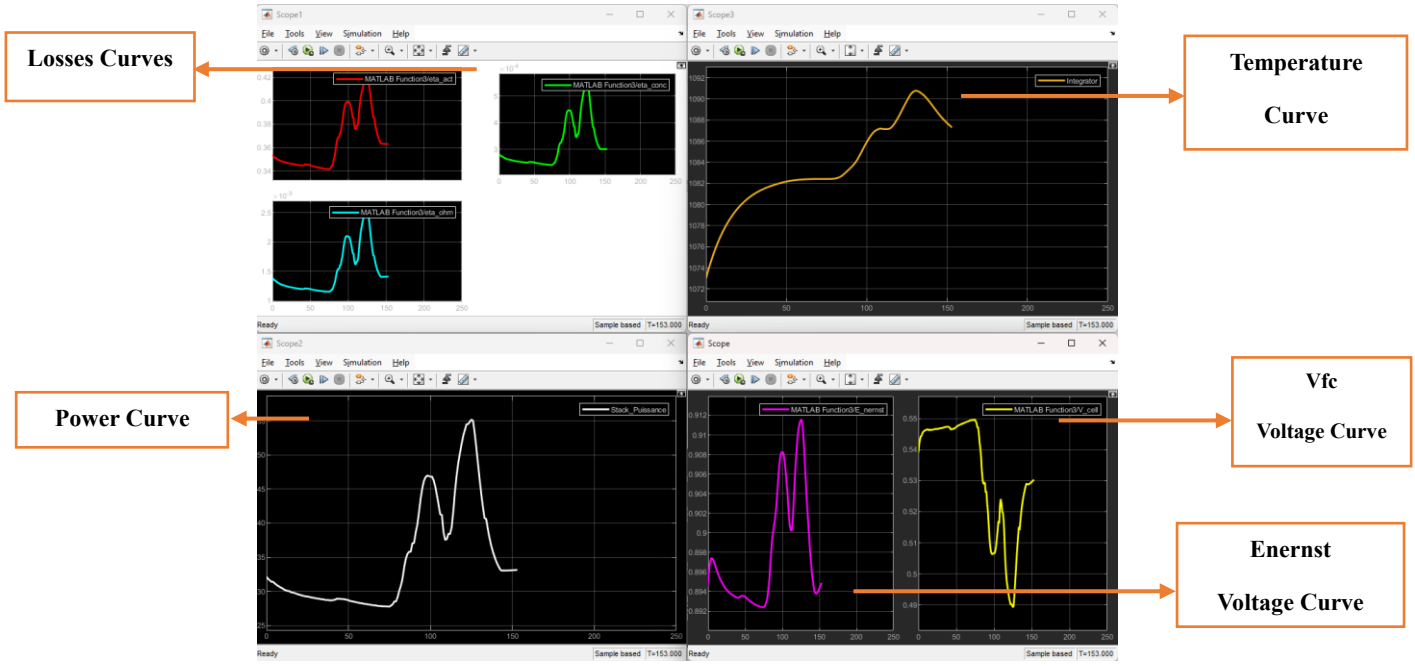


Figure IV. 14 Real-Time Simulation Results of the SOFC Digital Twin.

IV.5.2.6.1 Nernst Voltage and Fuel Cell Voltage

As the methane concentration rises, the Nernst voltage rises, this is because the amounts of gas participating in the electrochemical process have changed. Increased methane results in additional internal processes, which raise the theoretical voltage E_{nernst} because it depends on the partial pressures $P_{H_2}, P_{O_2}, P_{H_2O}$. The Nernst voltage also drops when CH_4 decreased. The Nernst E_{nernst} voltage is not precisely followed by the actual voltage V_{cell}. The real voltage may rise slightly at first when the CH_4 rises, but internal losses (activation, ohmic, and concentration) cause it to fall. With current, these losses rise. Losses are reduced and the voltage stabilizes when the CH_4 drops. The figure IV.15 illustrates the opposite behavior of the E_{nernst} and the V_{cell}.

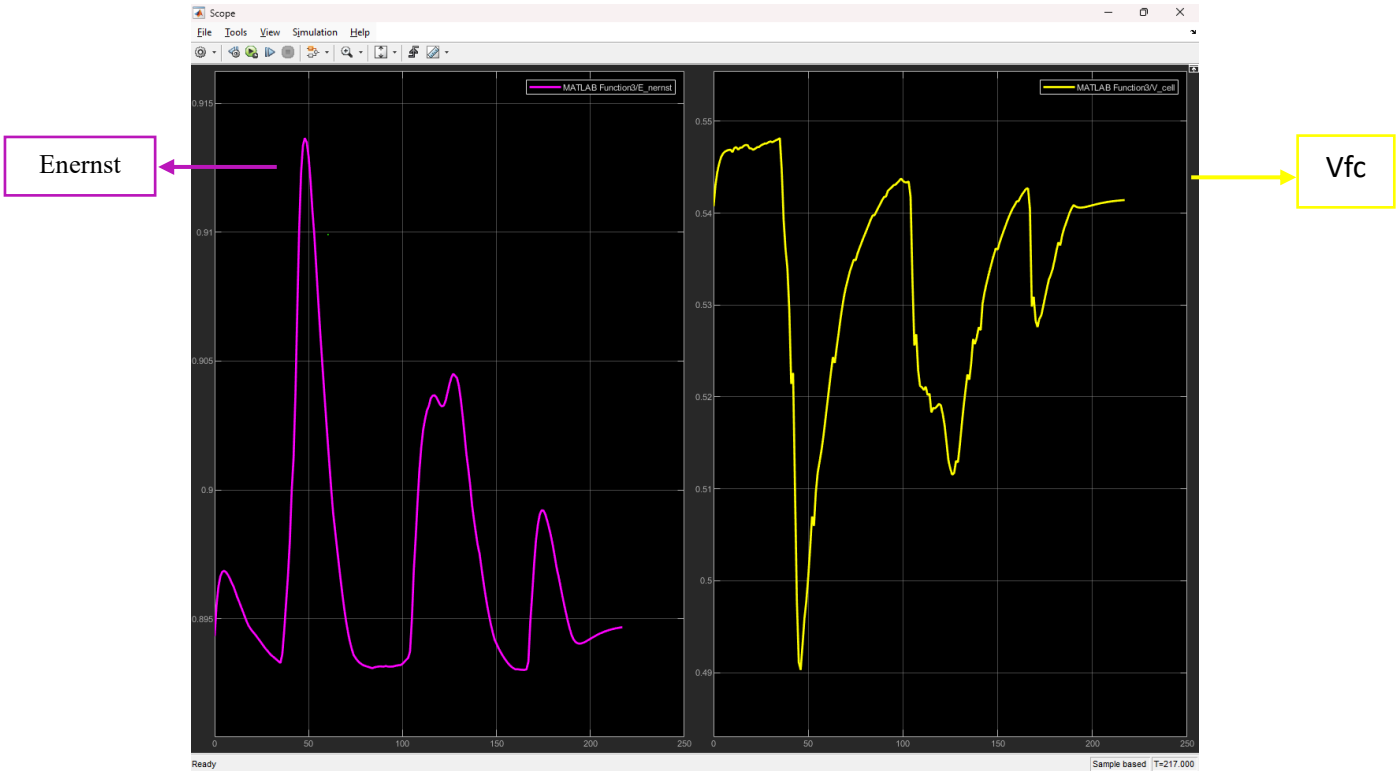


Figure IV. 15 The Behavior of the Enernst and Vfc curves.

IV.5.2.6.2 Activation, Concentration, and Ohmic losses

The behavior of curves of activation, concentration, and ohmic losses exhibit a similar trend throughout the real-time simulation. When the methane concentration increases, a larger amount of fuel becomes available for the electrochemical reactions, resulting in a higher current density. Consequently, all losses increase due to the intensified electrochemical activity inside the SOFC. When the methane concentration decreases, the current density decreases, leading to lower the activation, concentration, and ohmic losses. The figure IV.16 illustrates the behavior curves of activation, Concentration, and ohmic losses respectively.

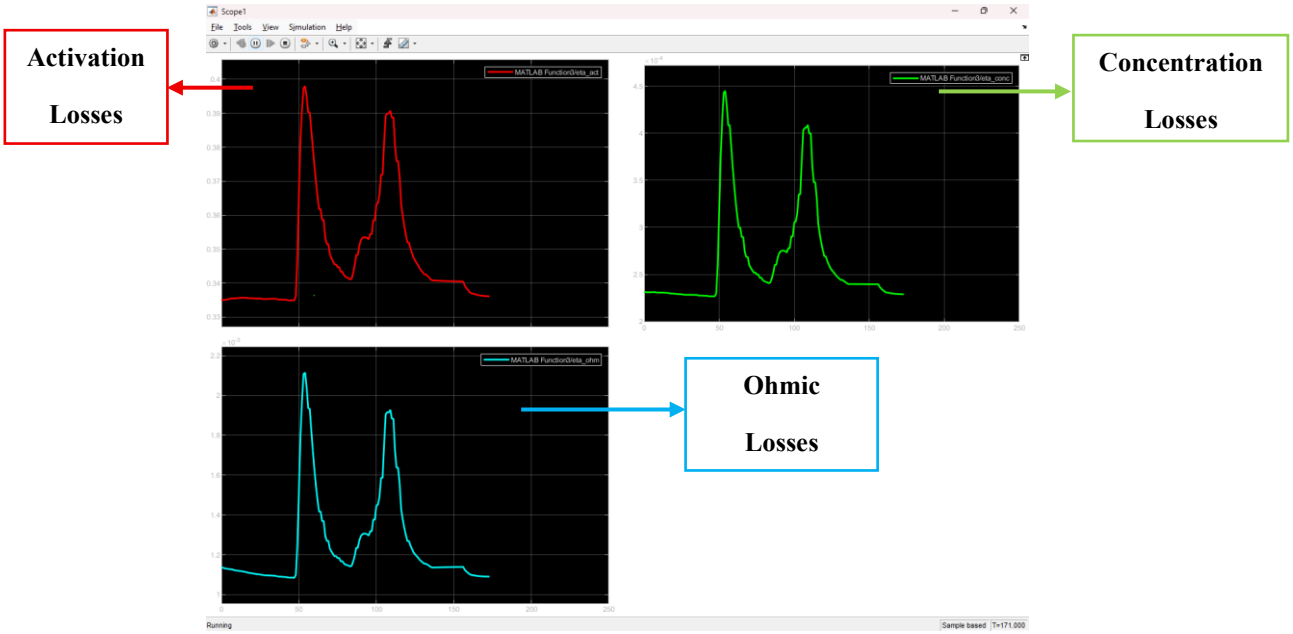


Figure IV. 16 The Behavior of the Activation, Concentration, and Ohmic Losses.

IV.5.2.6.3 Temperature

Because of the increased reactivity and heat output, the temperature rises as the concentration of methane increases. When the concentration of methane drops, less heat is produced. The temperature gradually stabilizes upon reaching the ambient air temperature. The figure IV.17 shows the temperature change curve in a real-time simulation.

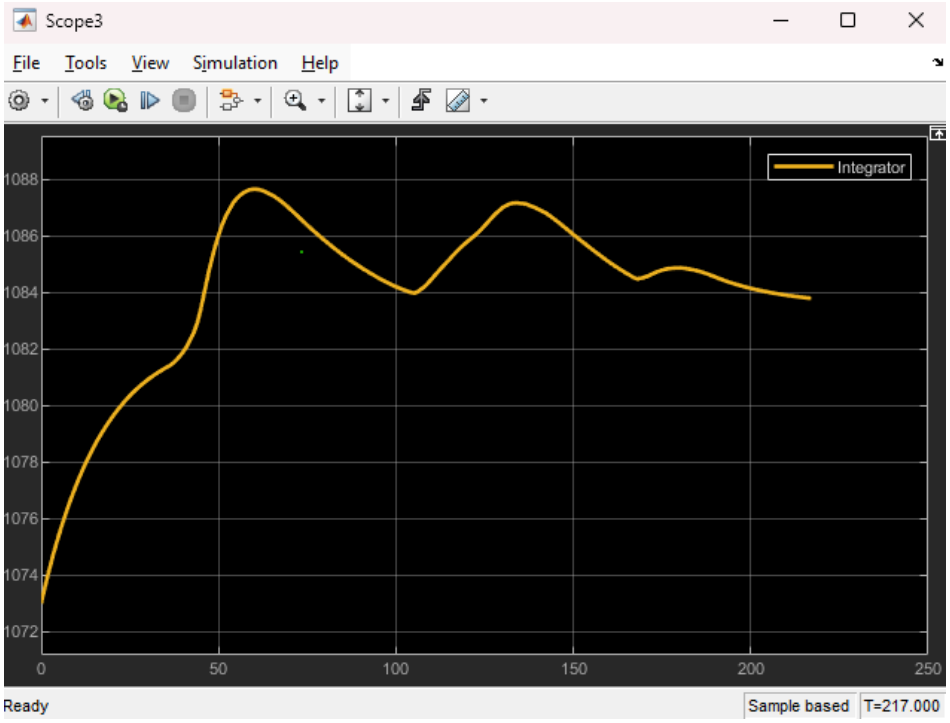


Figure IV. 17 The Temperature behavior curve of SOFC Model.

IV.5.2.6.4 Electrical Power

As the concentration of methane increases, energy output also increases, and this explains why the system produces more quantities of combustible fuel at higher methane concentrations, which in turn boosts hydrogen production and generates energy. The power output also increases due to its close relationship with potential. Conversely, energy output decreases significantly when the methane concentration decreases. The figure IV.18 illustrates the behavior of the power in our model.

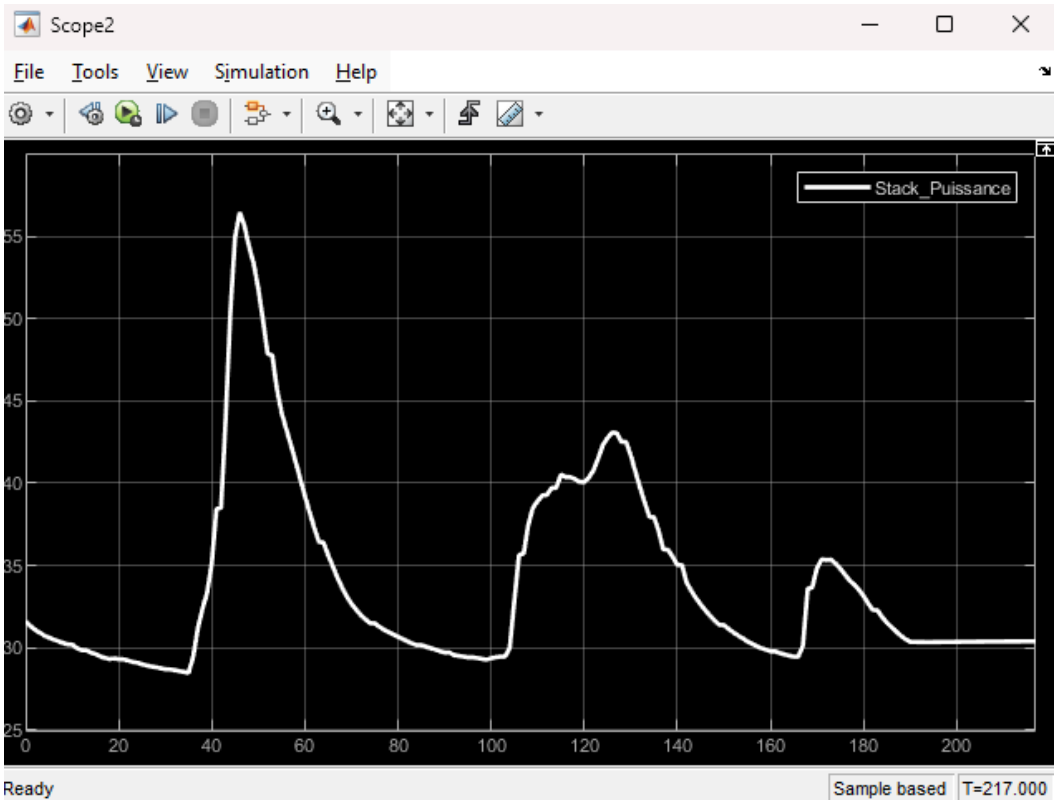


Figure IV. 18 The Electrical Power Behavior Curve of the SOFC Model.

IV.6 Future Work

The proposed proof-of-concept project demonstrated the feasibility of integrating a Solid Oxide Fuel Cell (SOFC) simulation model with methane sensing and IoT technology. In future work, the developed digital twin will be validated and extended using a real SOFC system. Real-time operational data, including temperature, voltage, current, and fuel flow rate, will be acquired directly from the physical fuel cell and synchronized with its digital counterpart. Operating under real-world conditions will enable more accurate performance monitoring, predictive analysis, fault diagnosis, and optimization of the fuel cell system. Furthermore, future developments will focus on enhancing the digital twin by incorporating advanced predictive

maintenance algorithms and artificial intelligence techniques to improve system reliability and operational efficiency.

IV.7 Conclusion

Based on methane concentration measurements, simulation results showed that the proposed system is capable of efficiently monitoring and analyzing the activity of a Solid Oxide Fuel Cell (SOFC) in real-time. Our digital twin model was validated by the observed changes in electrical and thermal parameters, which were consistent with the expected electrochemical behavior of the fuel cell. Also, this work demonstrates the feasibility of integrating digital twin technologies with IoT connectivity, and methane detection within a unified framework. The proposed digital twin prototype serves as a proof of concept for application and provides a solid foundation for further development, including actual fuel cell devices and more sophisticated monitoring and control applications.

General Conclusion

This project focused on developing an, IoT-based fuel cell monitoring system integrated with a digital prototype of a Solid Oxide Fuel Cell (SOFC). Our goal was to design an efficient solution that allows us to understand the behavior and performance of fuel cells in the presence of methane in real time before directly interacting with a real fuel cell.

Our proposed prototype combines a MQ-4 methane gas sensor, ESP32 microcontroller, MQTT communication protocol, and a digital model based on MATLAB/Simulink. This allows us to acquire, transmit, and process methane concentration data to simulate the electrical and thermal behavior of the solid oxide fuel cell. Our design enables real-time interaction between the built cell model and the monitoring system to analyze the cell performance.

The simulation and experimental results confirm the validity of the proposed system. The changes in methane concentration were successfully reflected in key fuel cell parameters, including Electrical Power, Nernst Voltage, Fuel Cell Voltage, Temperature, and Electrochemical Loss (Activation, Concentration, and Ohmic). The results demonstrate the ability of the digital twin concept that help to simulate the dynamic behavior of SOFC and confirm the feasibility of the proposed methodology.

We are gaining important insights into fuel cell modeling, IoT connectivity, and real-time data interchange through this study, which is our first hands-on experience with the notion of digital twin technology. In order to evaluate the Digital Twin under real-world operating conditions, a real fuel cell system can be included into the built prototype as future work. Improving the model's accuracy, putting predictive maintenance and defect detection strategies into practice, and creating a comprehensive industrial monitoring platform are possible future developments. The adoption of fuel cell technology and digital twin solutions for energy systems would be aided by these advancements.

REFERENCES

[1] International Energy Agency (IEA), «Global Methane Tracker 2025,» Paris, 2025.

[2] T. Lakshmi, P. Geethanjali et S. Krishna Prasad, «Mathematical modelling of solid oxide fuel cell using Matlab/Simulink,» chez *International Conference on Microelectronics, Communication and Renewable Energy (ICMiCR)*, Kanjirapally, India, 2013.

[3] H. Kagermann, W. Wahlster et J. Helbig, «Securing the future of German manufacturing industry: Recommendations for implementing the strategic initiative INDUSTRIE 4.0,» Acatech, Germany, München, 2013.

[4] K. Schwab, The Fourth Industrial Revolution, Geneva: World Economic Forum, 2016.

[5] D. Meraghni, L. Ouazzani, D. Hissel et N.Y. Steiner, «A data-driven digital-twin prognostics method for proton exchange membrane fuel cell remaining useful life prediction,» *International Journal of Hydrogen Energy,* vol. 46, pp. 2555-2564, 2021.

[6] M. Yue, K. Benaggoune, J. Meng et D. Diallo, «Implementation of an early stage fuel cell degradation prediction digital twin based on transfer learning,» *IEEE Transactions on Transportation Electrification,* 2021.

[7] M. Yue, Z. Li, R. Roche et N. Zerhouni, «Degradation identification and prognostics of proton exchange membrane fuel cell under dynamic load,» *Control Engineering Practice,* vol. 118, p. 104959, 2022.

[8] T. Testasecca, M. Picciotto Maniscalco, G. Brunaccini, G. Airò Farulla, G. Ciulla, M. Beccali et M. Ferraro , «Toward a Digital Twin of a Solid Oxide Fuel Cell Microcogenerator: Data-Driven Modelling,» *Energies,* vol. 17, n.116, p. 4140, 2024.

[9] Z. Guo, Z. Ye, P. Ni, C. Cao, X. Wei , J. Zhao et X. He, «Intelligent Digital Twin Modelling for Hybrid PV-SOFC Power Generation System,» *Energies ,* vol. 16, n.16, p. 2806, 2023.

[10] H. B. e. al., «Solid Oxide Fuel Cell Voltage Prediction by a data-driven Approach,» *Energies,* vol. 18, n.19, 2025.

[11] Algérie Confluences, «Sonatrach s'engage pour le climat : une feuille de route ambitieuse pour réduire les gaz torchés à 1 % d'ici 2030,» 24 June 2025. [En ligne]. Available: https://algerieconfluences.dz/sonatrach-sengage-pour-le-climatune-feuille-de-route-ambitieuse-pour-reduire-les-gaz-torches-a-1-dici-2030/. [Accès le 11 April 2026].

[12] M. Boudellal, l'hydrogéne et ses applications, 2e éd, Paris: Dunod, 2012.

[13] J. Lachaize, «Etude des stratégies et des structures de commande pour le pilotage des systèmes énergétiques à pile à combustible (PAC) destinés à la traction,» 2004.

[14] F. Barbir, PEM Fuel Cells(Theory and Practice), Academic Press, 2012.

[15] R. O'Hayre, S.-W. Cha, W. Colella et B.F. Prinz, Fuel Cell Fundamentals, 2nd edition éd., J. W. &. Sons, Éd., Hoboken, NJ, 2009.

[16] Laurencin et Jérôme, «Fonctionnement sous methane d'une pile à combustible "SOFC": optimisation des performances et de la durabilité,» Grenoble, 2009.

[17] «Manufacturing and characterization of single cell intermediate-temperature solid oxide fuel cells for APU in transportation application,» [En ligne]. Available: https://www.researchgate.net/figure/Planar-SOFC-design-21-The-Figure-I12-shows-different-parts-of-planar-SOFC-necssary-to_fig10_281185027. [Accès le 29 April 2026].

[18] J. Padullés, G.W. Ault et J.R. McDonald, «An integrated SOFC plant dynamic model for power systems simulation,» *Journal of Power Sources,* vol. 86, pp. 495-500, March 2000.

[19] A. Gelen et T. Yalcinoz, «The performance of thermal based modified solid oxide fuel cell (SOFC) model under different DC load conditions,» *Journal of Engineering Reasearch,* vol. 3, n.13, p. p. 23, 2015.

[20] A. Gebregergis, P. Pillay, D. Bhattacharyya et R. Rengaswamy, «Solid oxide fuel cell modeling,» *IEEE Transactions on Industrial Electronics,* vol. 56, n.11, pp. 139-148, 2009.

[21] M. Khan et M.T. Iqbal, «Dynamic modeling and simulation of a fuel cell generator,» *Fuel Cells,* vol. 5, n.11, pp. 97-104, 2005.

[22] A. Brabie et W. Hasselbring, «From Digital Twins to Digital Twin Prototypes: Concepts, Formalization, and Applications,» *IEEE Access,* vol. 12, pp. 75337-75365, 2024.

[23] S. Saadabadi, T. Woudstra et P.V. Aravind, «Direct internal methane reforming in biogas fuelled solid oxide fuel cell: The influence of operating parameters,,» *Energy Science & Engineering,* vol. 9, n.18, pp. 1232-1248, 2021.

[24] C. Lin, F. Kerscher, B. Steinrücken, S. Ohmstedt et H. Spliethoff, «Investigation on cell-to-cell variations within methane-rich internal reforming solid oxide fuel cell (SOFC) stacks,,» *Chemical Engineering Journal,* vol. 526, 2025.

[25] F. Tao, H. Zhang, A. Liu et A. Nee, «Digital twin in idustry: State-of-the-art,» *IEEE Transactions on Industrial Informa,* vol. 15, n.14, pp. 2405-2415, April 2019.

[26] J. Uri, «50 Years Ago: Apollo 13 Review Board Report,» June 2020. [En ligne]. Available: https://www.nasa.gov/history/50-years-ago-apollo-13-review-board-report/. [Accès le Sunday April 2026].

[27] M. Grieves, «Digital twin: Manufacturing excellence through virtual factory replication,» 2014.

[28] E. Glaessgen et D.S. Stargel, «The digital twin paradigm for future NASA and U.S. Air Force vehicles,» chez *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conf.*, Honolulu, HI, USA, 2012.

[29] Gartner, «GARTNER'S TOP STRATEGIC TECHNOLOGY TRENDS,» 2017.

[30] A. Rasheed, O. San et T. Kvamsdal, «Digital twin: Values, challenges and enablers from a modeling perspective,» *IEEE Access,* vol. 8, pp. 21980-22012, 2020.

[31] M. Shafto, «Modeling, simulation, information technology and processing roadmap,» n° %111, 2010.

[32] M. Grieves et J. Vickers, «Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems,» chez *Transdisciplinary Perspectives on Complex Systems*, F. Kahlen, S. Flumerfelt et A. Alves, Éds., Cham , Springer, 2017, pp. 85-113.

[33] T. Uhlemann, C. Lehmann et R. Steinhilper, «The digital twin: Realizing the cyber-physical production system for Industry 4.0,» *Procedia CIRP,* vol. 61, pp. 335-340, 2017.

[34] D. M. Botín-Sanabria, A. S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza et J. D. Lozoya-Santos, «Digital twin technology challenges and applications: A comprehensive review,» *Remote sensing,* vol. 14, n.16, 2022.

[35] W. Kritzinger, M. Karner, G. Traar, J. Henjes et W. Sihn, «Digital twin in manufacturing: A categorical literature review and classification,» *IFAC-PapersOnline,* vol. 51, n° %111, pp. 1016-1022, 2018.

[36] G. Caiza et R. Sanz, «Digital twin to control and monitor an industrial cyber-physical environment supported by augmented reality,» *Applied Sciences,* vol. 13, n.113, 2023.

[37] A. Chai, Y. Yin , M. Lian, Y. Sun, C. Guo, L. Wang et Z. Fang, «DUA-MQTT: A distributed high-availability message communication model for the industrial Internet of Things,» *Sensors,* vol. 25, n.116, 2025.

[38] M. R. e. al., «Industry 4.0: The future of productivity and growth in manufacturing industries,» Boston Consulting Group, 2015.

[39] Q. QI, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang et A.Y.C. Nee, «Enabling technologies and tools for digital twin,» *Journal of Manufacturing Systems,* vol. 58, pp. 3-21, 2021.

[40] Siemens AG, «MindSphere: The industrial IoT as a service solution,» Siemens AG, Munich, 2020.

[41] K. Ashton, «That "Internet of Things" thing,» *RIFD Journal,* vol. 22, n.17, pp. 97-114, 2009.

[42] IoT Analytics, «State of IoT 2023: Number of connected IoT devices growing 16% to 16.7 billion globally,» IoT Analytics, Hamburg, 2023.

[43] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang et W. Zhao, «A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications,» *IEEE Internet of Things Journal,* vol. 4, n.15, pp. 1125-1142, 2017.

[44] Ansys, «Ansys Twin Builder | Create and Deploy Digital Twin Models,» Ansys, [En ligne]. Available: Ansys Twin Builder | Create and Deploy Digital Twin Models. [Accès le 21 May 2026].

[45] S. D. I. Software, «Digital Twin,» Siemens Digital Industries Software, [En ligne]. Available: https://www.siemens.com/en-us/technology/digital-twin/. [Accès le 21 May 2026].

[46] MatWorks, MATLAB and Simulink User's Guide, Natick, MA: The MathWorks, Inc., 2023.

[47] MathWorks, «Simulink Real-Time,» 2023. [En ligne]. Available: https://www.mathworks.com/products/simulink-real-time.html. [Accès le 30 April 2026].

[48] N. Naik, «Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP,» chez *IEEE International Systems Engineering Symposium (ISSE)*, Vienna, 2017.

[49] H. GmbH, «MQTT essential: A lightweight IoT protocol,» HiveMQ, Landshut, 2022.

[50] A. Banks et R. Gupta, «MQTT Version 3.1.1,» 2014. [En ligne]. Available: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html. [Accès le 30 April 2026].

[51] V. Karagiannis, P. Chatzimisios, F. Vázquez-Gallego et J. Alonso-Zárate, «A Survey on Application Layer Protocols for the Internet of Things,» *Transaction on IoT and Cloud Computing,* vol. 3, n.11, pp. 9-18, 2015.

[52] S. Meraghni, L. Terrissa, M. Yue, J. MA, S. Jemei et N. Zerhouni, «A data-driven digital-twin prognostics method for proton exchange membrane fuel cell remaining useful life prediction,» *International Journal of Hydrogen Energy,* vol. 46, n.12, pp. 2555-2564, 2021.

[53] M. Zhang, Y. Xu et A. Amiri, «Digital-Twin Enabled online Remaining Useful Life Predicition of PEM Fuel Cell,» chez *IEEE Ineternational Conference on Prognostics and Health Management (ICPHM)*, 2023.

[54] L. Chalal, Allal Saadane et Ahmed Rachid, «Unified Environment for Real Time Control of Hybrid Energy System Using Digital Twin and IoT Approach,» *Sensors,* vol. 23, n.112, p. 5646, 2023.

[55] R. Huda, A. Winarno, P.S. Bagaskara, M.I. Anwari, A.A. Ismail, M.H.M. Ariff et I. Bahiuddin, «Design of Digital Twin for Movement Simulation and Anomaly Analysis of a Forklift Lifting Mechanism Based on IoT,» chez *International Conference on Sustainable Energy: Toward Energy Transition and Net-Zero Emission (ICOSE 2025)*, 2025.

[56] L. Hanwei Electronics Co., «Flammable Gas Sensor (Model:MQ-4) Manual Version: 1.3,» 2014.

[57] L. Hanwei Electronics Co., «MQ-4 Gas Sensor Technical Data».

[58] Espressif Systems, «ESP32 Series Datasheet, version 5.2,» 2025. [En ligne]. Available: https://documentation.espressif.com/esp32_datasheet_en.pdf. [Accès le 21 May 2026].

[59] E. Systems, «ESP Product Selector,» Espressif Systems, [En ligne]. Available: https://products.espressif.com/#/product-selector?names=&filter=. [Accès le 24 May 2026].

[60] K. Oliynyk, «ESP32 Development Platforms: Detailed Analysis & Boards Comparison,» WebbyLab, 28 April 2025. [En ligne]. Available: https://webbylab.com/blog/esp32-boards-comparison/. [Accès le 24 May 2026].

[61] R. Mischianti, «ESP32 NODEMCU-32S ESP-32S Kit: high resolution pinout, datasheet, and specs,» Renzo Mischianti, 1 February 2024. [En ligne]. Available: https://mischianti.org/esp32-nodemcu-32s-esp-32s-kit-high-resolution-pinout-datasheet-and-specs/. [Accès le 24 May 2026].

[62] R. Bishop, Mechatronics: An Introduction, Boca Raton, Florida: CRC Press, 2005.

[63] Electrónicos Caldas, «LTE12 Series Active Buzzer Datasheet,» Electrónicos Caldas, Colombia.

[64] IITian Academy, «Edexcel iGCSE Physics - 2.11–2.12 Resistance of LDRs and Thermistors; Lamps and LEDs - Study Notes,» IITian Academy, 11 February 2026. [En ligne]. Available: https://www.iitianacademy.com/edexcel-igcse/edexcel-igcse-physics-4ph1-exam-style-questions-and-study-notes/edexcel-igcse-physics-2-11-2-12-resistance-of-ldrs-and-thermistors-lamps-and-leds-study-notes/. [Accès le 22 May 2026].

[65] N. Poole, «Light-Emitting Diodes (LEDs),» SparkFun Electronics, 2013. [En ligne]. Available: https://learn.sparkfun.com/tutorials/light-emitting-diodes-leds/all. [Accès le 22 May 2026].

[66] K. Sedghisigarchi et A. Feliachi, «Dynamic and transient analysis of power distribution systems with fuel cells-part I: fuel-cell dynamic model,» *IEEE Transactions on Energy Conversion,* vol. 19, n.12, pp. 423-428, 2004.

[67] Y. Liu, J. Liu, L. Fu et Q. Wang, «Numerical study on effects of flow channel length on solid oxide fuel cell-integrated system performances,» *Sustainability,* vol. 16, n.14, p. 1643, February 2024.

[68] K. Chahar, A.K. Maurya et A. Chaturved, «Performance analysis of fuel cell assisted distributed generation system,» chez *International Conference on Contemporary Computing and Informatics (IC3I)*, 2014.

[69] MathWorks, «Discrete- Time Integrator,» 2025. [En ligne]. Available: https://www.mathworks.com/help/simulink/slref/discretetimeintegrator.html. [Accès le 05 May 2026].

[70] Fuel Cell Store, «Determining the Number of Cells for your Hydrogen/Air Fuel Cells,» Fuel Cell Store, 2021. [En ligne]. Available: https://www.fuelcellstore.com/blog-section/fuel-cell-information/hydrogen-air-calculator. [Accès le 16 May 2026].

[71] S.Thiers, *Cours de piles à combustible,* 2009.

[72] S. Bouchaala et H. Beggari, «Etude d'un système de pile à combustible de type PEMFC,» 2022.

[73] L. Zhang, H.Q. He, W.R. Kwek, J. Ma, E.H. Tang et S.P. Jiang, «Fabrication and characterization of anode-supported tubular solid-oxide fuel cells by slip casting and dip coating techniques,» *Journal of the American Ceramic Society,* vol. 92, pp. 302-310, 2009.

[74] P. Halder, M. Barbaie, F. Salek, K. Shah, S. Stevanovic, T.A. Bodisco et A. Zare, «Performance, emissions and economic analyses of hydrogen fuel cell vehicles,» *Renewable and Sustainable Energy Reviews,* vol. 199, p. 114543, 2024.

[75] L. Atzori, A. Lera et G. Morabito, «The internet of things: A survey,» *Computer Networks,* vol. 54, pp. 2787-2805, October 2021.

[76] N. Instruments, «Sensor Fundamentals,» National Instruments, [En ligne]. Available: https://www.ni.com/en/shop/data-acquisition/sensor-fundamentals.html. [Accès le 22 May 2026].

[77] Monolithic Power System, «What is a sensor?,» Monolithic Power System, [En ligne]. Available: https://www.monolithicpower.com/en/learning/mpscholar/sensors/intro-to-sensors/what-is-a-sensor. [Accès le 22 May 2026].

[78] A. Prabhu, «How MQ2 Gas/Smoke Sensor Works? & Interface it with Arduino,» Last Minute Engineers, 20 January 2026. [En ligne]. Available: https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/. [Accès le 22 May 2026].

[79] N. Jain, A. Roy, R. Jain et N. Karmakar, «Polymer Electrolyte Membrane Fuel Cells: Alternative to fossil fuels for power supply to heavy earth moving and allied machinery in mining and civil engineering industry,» 2013.