**System Requirements  
Specification (SRS)  
  
PV wireless monitoring**

|  |
| --- |
| Using this Document – System Integration Benefits |
| The System Requirements Specification (SRS) document describes all data, functional and behavioral requirements of the hardware and software under production or development. |

**Table of Contents**

[1. Introduction 3](#_Toc475354427)

[1.1. Purpose 3](#_Toc475354428)

[1.2. Scope 3](#_Toc475354429)

[1.3. Context 3](#_Toc475354430)

[1.4. Organization 4](#_Toc475354431)

[3. Overall Description 6](#_Toc475354432)

[3.1. Product perspective 6](#_Toc475354433)

[3.2. Product features 7](#_Toc475354434)

[3.3. User classes and characteristics 7](#_Toc475354435)

[3.4. Operating environment 7](#_Toc475354436)

[3.5. Design and implementation constraints 7](#_Toc475354437)

[3.6. User documentation requirements 7](#_Toc475354438)

[3.7. Assumptions and dependencies 8](#_Toc475354439)

[3.8 System requirements 8](#_Toc475354440)

[4. Specific Requirements 9](#_Toc475354441)

[4.1. Operations 9](#_Toc475354442)

[4.2. Communication 9](#_Toc475354443)

[4.3. Safety/reliability 9](#_Toc475354444)

[5. Data Model and Description 10](#_Toc475354445)

[5.1. Use Case Diagram 10](#_Toc475354446)

[5.2. Domain Model 13](#_Toc475354447)

[5.3. Sequence Diagrams 17](#_Toc475354448)

[5.4. State Diagram 18](#_Toc475354449)

[5.5. Prototype 19](#_Toc475354450)

[Prototype Operation 19](#_Toc475354451)

[Sample Scenario 19](#_Toc475354452)

[6. References 20](#_Toc475354453)

# Introduction

This section presents an overview of the SRS to help the reader understand how the document is organized and how to read and interpret it. The System Requirements Specification (SRS) document describes all data, functional and behavioral requirements of the hardware and software.

Renewable Energy System development are inherent to the power electronics development. In the scope of the microgeneration project of SYSTEC-ENERGY Research Unit – Smart Energy Systems and Technologies – a distributed PV harvesting with multiple power converters is in development. Despite of the standalone behavior of the PV power converters, a monitoring subsystem is expected to be implemented.

## Purpose

The purpose of this document is to describe the specific requirements for the Wireless PV monitoring system. This document is intended to provide an unambiguous, concise, and complete list of requirements to help design the Gateway and the nodes attached to each PV power converter. This document will include constraints and show how to use the system.

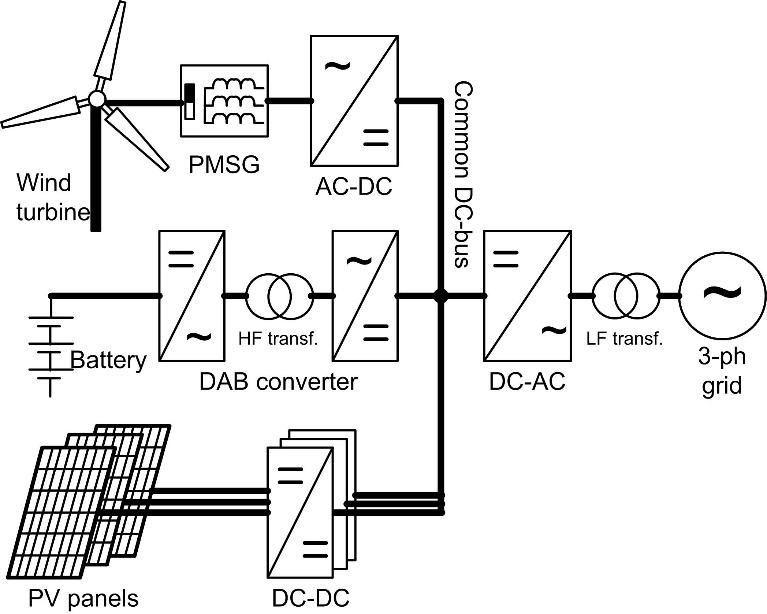
## Scope

The Wireless PV monitoring system is a system intended to acquire the data and behavior status of each PV power converter.

## Context

Place the software in a business or product line context with the aim for the reader to understand the 'big picture'. Strategic issues relevant to context are discussed.

This work is inserted in the scope of a microgeneration project of the SYSTEC-ENERGY Research Unit. Figure 1 presents the main architecture of this project. Currently, several work has been done in this project to implement a monitoring subsystem. This work focuses on data collection from each power converter and also on its control by sending references and actions. At the moment of this proposal, no work has been done to implement the monitoring feature in the PV converters. This way, a wireless network implementation is proposed to monitor the PV power converters.



*Figure 1 - Main architecture of microgeneration project.*

The PV converter uses a non-isolated high gain DC/DC topology and has a PIC32 microcontroller which implements a MPPT algorithm in the control loop. It is currently possible to connect a device with wireless capabilities to the PV power converter. The converter is able to provide power and data. The data collected from the monitoring device is sent to an aggregator node for post processing and data storage. Currently, several work has already been done to have a reliable wired data aggregator/concentrator. This subsystem is also responsible for sending user commands to the power converters and presenting the user interface in a webpage remotely accessed via a web browser.

## Organization

The rest of the document is broken down into six sections.

Section 2 contains the overall description. In this section, the document will provide information about the system functions. Also included in this section is a description of the constraints needed for safety, as well as any assumptions made in the user characteristics and dependencies.

Section 3 has a hierarchical enumerated list of the requirements the system must fulfill.

Section 4 contains various models that specify the bridge between application domain and machine domain. In this section, UML diagrams and their respective detailed explanations are used to visualize and demonstrate the expected behavior of the system.

Section 5 gives an overview of the prototype, describes how to access it and run it, as well as examples of sample scenarios.

Section 6 gives a list of all the references used to compile this document.

Section 7 gives a point of contact if more information is required.

# OVERALL DESCRIPTION

Present here a high-level overview of the product being specified and the environment in which it will be used, the anticipated users of the product, and the known constraints, assumptions, and dependencies.

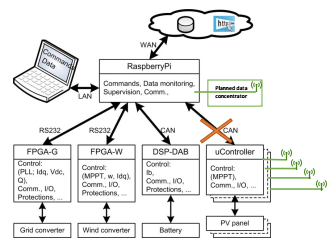
The system to be developed is a wireless monitoring system capable of acquiring data from several PV converters, supporting the mechanical and EMI influences from the environment and being integrated to the current monitoring infrastructure.

## Product perspective

Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.

The wireless feature is intended to be added to the monitoring infrastructure supported in a System-on-Chip computer – namely a Raspberry Pi 2 – and the PV power converters.

The diagram in the following figure presents the overall perspective of the development expected by this SRS



The major requirements of the microgeneration monitoring system are the data acquisition and storage, having a website interface that might be easily accessed from the internet (with all the security constrains guaranteed). The relevant data must be identified, collected and transmitted using regular communication channels (RS232, SPI, CAN, etc).

## Product features

Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or a class diagram, is often effective.

This addition to the microgeneration project has two main objectives:

1. Extend the monitoring feature to the PV power converters
2. Implement and demonstrate a feasible wireless communication between the PV power converters and the head of the system

## User classes and characteristics

Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.

No user classes are expected in this communication module. The user interaction is defined by the upper layer of this system.

## Operating environment

Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.

This system will be implemented in a cabinet with EMI constrains caused by the power converters. This SRS will include the specification of the hardware platform.

## Design and implementation constraints

Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).

## User documentation requirements

List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.

## Assumptions and dependencies

List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change.

Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).

## Minimum system requirements

Describe the minimum system requirements for this product (e.g. platform version, disk space, RAM, etc.)

# Specific Requirements

## Operations

## The system must collect data from more than 1 PV converter (up to 10 PV converters)

## Each wireless node should be powered by 5V

## The wireless concentrator should be powered by 5V

## The system concentrator node must have an accessible broadcast mode that provides the received raw data from remote nodes.

## Communication

## Each wireless node should have a serial communication

## The wireless concentrator should have a serial communication

## The wireless concentrator should implement the microgen serial protocol

## A wireless concentrator must collect data from all the nodes in less than 10 seg

## All wireless nodes must send data to the wireless concentrator upon a request.

## Safety/reliability

## All data shall be sampled and hold upon a SYNC command.

## The data jitter must be .

## Requirements analysis

The project shall perform the tasks of requirements analysis for the purpose of establishing what the system will be capable of accomplishing; how well system products are to perform in quantitative, measurable terms; the environments in which system products operate; the requirements of the human/system interfaces; the physical/aesthetic characteristics; and constraints that affect design solutions. The market needs, requirements, and constraints are derived from stakeholder expectations, project and enterprise constraints, external constraints, and higher-level system requirements. These are documented in a requirements baseline. The requirements baseline guides the remaining activities of the SEP and represents the definition of the problem to be solved. The tasks associated with requirements analysis are identified in Figure 10. The project assesses and analyzes inputs defined in tasks 6.1.1 through 6.1.9 to identify cost, schedule, and performance risks; to define functional and performance requirements; and identify conflicts. Trade-off analyses are conducted to resolve such conflicts so as to arrive at a balanced requirements baseline. The trade-off analysis and risk assessment, analysis, and handling tasks are discussed in 6.7. For each application of the SEP, the project refines previously defined requirements for upper levels of the system architecture, as appropriate, and defines requirements for the system under development (refer to the discussion in 1.3). There are a large number of requirements that the wireless PV monitoring must meet. There requirements can be split up into requirements for operation, communication and safety/reliability of the data, as detailed below. [from IEEE\_STD 1220\_2005]

## Define stakeholder expectations

The project defines and quantifies the stakeholder expectations for the system. Stakeholder expectations may come from marketing, an acquirer’s order, a recognized market opportunity, direct communications from users, or the requirements from a higher-level system. Stakeholder expectations include the following:

a) What the stakeholder wants the system [product(s), life cycle processes, and desired quality factors] to accomplish (functional requirements)

b) How well each function is to be accomplished (performance requirements)

c) The natural and induced environments in which the product(s) of the system operates or may be used

d) Constraints (e.g., funding; cost or price objectives; schedule; technology; nondevelopmental and reusable items; design characteristics; hours of operation per day; on-off sequences; external interfaces; and specified existing equipment, procedures, or facilities related to life cycle processes)

The definition of stakeholder expectations is balanced with an analysis of the effects on the overall system design and performance, as well as on the human engineering; knowledge, skills, and abilities; availability; reliability; safety; and training requirements of the humans required to support life cycle processes.

* 1. Define project and enterprise constraints

The enterprise identifies and defines project and enterprise constraints that impact design solutions.

* + 1. Project constraints

Project constraints may include the following:

a) Approved specifications and baselines developed from prior applications of the SEP

b) Engineering and technical plans

c) Team assignments and structure

d) Automated tools availability or approval for use

e) Control mechanisms

f) Required metrics for measuring technical progress

g) Reuse and commercial-off-the-shelf (COTS)

* + 1. Enterprise constraints

Enterprise constraints may include the following:

a) Management decisions from a preceding technical review

b) Enterprise general specifications, standards, or guidelines

c) Policies and procedures

d) Domain technologies

e) Established life cycle process capabilities

f) Physical, financial, and human resource allocations to the technical effort

The enterprise may also impose long-range planning constraints on the technical effort, which may demand an evolutionary development strategy in order to achieve project or enterprise goals.

* + 1. Define external constraints

The project identifies and defines external constraints that impact design solutions or the implementation of SEP activities. These constraints include the following:

a) Public and international laws and regulations

b) The technology base

c) Industry, international, and other general specifications, standards, and guidelines

d) Human-related specifications, standards, and guidelines

e) Human availability, recruitment, and selection

f) Competitor product capabilities

* + 1. Define operational scenarios

The project identifies and defines the operational scenarios that define the range of the anticipated uses of system product(s). For each operational scenario, the project defines expected interactions with the environment and other systems; human tasks and task sequences; and physical interconnections with interfacing systems, platforms, or products.

* + 1. Define measures of effectiveness

The project defines system effectiveness measures that reflect overall stakeholder expectations and satisfaction. Key MOEs may include performance, safety, operability, usability, reliability, maintainability, time and cost to train, workload, human performance requirements, or other factors.

* + 1. Define system boundaries

The project defines the following:

a) Which system elements are under design control of the project and which fall outside their control

b) The expected interactions among system elements under design control and external and/or higherlevel and interacting systems outside the system boundary

* + 1. Define interfaces

The project defines the functional and design interfaces to external and/or higher-level and interacting systems, platforms, humans, and/or products in quantitative terms. Mechanical, electrical, thermal, data, communication-procedural, human-machine, and other interactions are included.

* + 1. Define utilization environments

The project defines the utilization environments for each of the operational scenarios. All environmental factors, natural or induced, that may affect system performance are identified and defined. Factors that ensure that the system minimizes the potential for human or machine errors or failures that cause injurious accidents or death, and impart minimal risk of death, injury, or acute chronic illness, disability, and/or reduced job performance of the humans who support the system life cycle are identified. Specifically, weather conditions (e.g., rain, snow, sun, wind, ice, dust, and fog), temperature ranges, topologies (e.g., ocean, mountains, deserts, plains, and vegetation), biological (e.g., animal, insects, birds, and fungi), time (e.g., day, night, and dusk), induced (e.g., vibration, electromagnetic, acoustic, and chemical), or other environmental factors are defined for possible locations and conditions where the system may be operated. Effects on hardware, software, and humans are assessed for impact on system performance and life cycle processes.

* + 1. Define life cycle process concepts

The project analyzes the outputs of tasks 6.1.1 through 6.1.8 to define life cycle process requirements necessary to develop, produce, test, distribute, operate, support, train, and dispose of system products under development.

* + - 1. Manpower

The project identifies and defines the required job tasks and associated workload used to determine the number and mix of humans who support the system life cycle processes.

* + - 1. Personnel

The project evaluates and determines the human experiences, aptitudes, knowledge, skills, and abilities required to perform the job tasks that are associated with the humans who support the system life cycle processes.

* + - 1. Training

The project identifies and develops the instruction, education, and on-the-job or team training necessary to provide humans and teams with knowledge and job skills needed to support the system life cycle processes at the specified levels of performance. This includes the tools, devices (including embedded training systems), training simulators, techniques, procedures, and training materials and technical manuals to be developed and employed to provide training for all required tasks.

* + - 1. Human engineering

The project identifies and accounts for those human cognitive, physical, and sensory characteristics of the humans who support the system life cycle that directly contribute to, or constrain, system performance and that impact human-machine interfaces.

* + - 1. Safety

The project accounts for the system design features that create significant risks of death, injury, or acute chronic illness, disability, and/or reduce job performance of personnel who operate, maintain, or support the system.

* + 1. Define functional requirements

The project performs functional context analysis (see 6.3.1) for the purpose of defining what the system should be able to do (the functional requirements). The functions identified are used in 6.1.11 to define how well the functions must be performed and to establish the performance requirements. The functions identified through functional context analysis are further decomposed during functional decomposition (see 6.3.2) to provide a basis for identifying and assessing design alternatives. All requirements of a system typically involve a functional and performance aspect, and it is viewing system requirements as having both functional and performance aspects that ensures that the requirements are complete, consistent, and verifiable.

* + 1. Define performance requirements

The project defines the performance requirements for each function of the system. Performance requirements describe how well functional requirements must be performed to satisfy the MOEs. These performance requirements are the MOPs that are allocated to subfunctions during functional decomposition analysis and that are the criteria against which design solutions [derived from synthesis (see 6.5)] are measured. There are typically several MOPs for each MOE, which bind the acceptable performance envelope.

* + 1. Define modes of operation

The project defines the various modes of operation (embedded training capability, fully operational, etc.) for the system products under development. The conditions (environmental, configuration, operational, etc.), which determine the modes of operation, are defined.

* + 1. Define technical performance measures

The project identifies the technical performance measures (TPMs), which are key indicators of system performance. Selection of TPMs are usually limited to critical MOPs that, if not met, put the project at cost, schedule, or performance risk. Specific TPM activities are integrated into the SEMS to periodically determine achievement to date and to measure progress against a planned value profile.

* + 1. Define design characteristics

The project identifies and defines required design characteristics (e.g., color, texture, size, anthropomorphic limitations, weight, and buoyancy) for the system products under development. The project identifies which design characteristics are constraints and which can be changed based on trade-off analyses.

* + 1. Define human factors

The project identifies and defines human-factor considerations (e.g., design space limits, climatic limits, eye movement, reach, ergonomics, cognitive limits, and usability) that affect operation of the system under development. The project identifies which human factors are constraints and which can be changed based on trade-off analyses.

* + 1. Establish requirements baseline

The output of tasks 6.1.1 through 6.1.15 is recorded in three views (operational, functional, and design) to form a requirements baseline that establishes the system problem to be solved by the project. The operational view describes how the system products serve their users. It establishes who operates and supports the system and its life cycle processes, and how well and under what conditions the system products are to be used. The functional view describes what the system products do to produce the desired behavior described in the operational view and provides a description of the methodology used to develop the view and decision rationale. The design view describes the design considerations of the system products development and establishes requirements for technologies and for design interfaces among equipment and among humans and equipment. The content of these views may include the following:

a) Operational view

1) Operational need description

2) Results of system operational analyses

3) Operational sequences/scenarios (best portrayed in pictures), which include utilization environments, MOEs, and how the system products should be used

4) Conditions/events to which system products should respond

5) Operational constraints, including MOEs

6) Identified human roles, including job tasks and skill requirements

7) Training requirements, including how humans may be trained to be a part of the system and

support system life cycle processes through formal, informal, embedded, on-the-job, or other

forms of training

8) Identification of what operations are required to ensure safety

9) Life cycle process concepts to include MOEs, critical MOPs, and already existing products and

services

10) Operational interfaces with other systems, platforms, humans, and/or products

11) System boundaries

b) Functional view

1) Functional requirements that describe what system products and life cycle processes must do or

Accomplish

2) Performance requirements including qualitative (how well), quantitative (how much, capacity),

and time lines or periodicity (how long, how often) requirements

3) Functional sequences for accomplishing system objectives

4) TPM criteria

5) Functional interface requirements with external, higher-level, or interacting systems, platforms,

humans, and/or products

6) Modes of operations

7) Functional capabilities for planned evolutionary growth

c) Design view

1) Previously approved specifications and baselines

2) Design interfaces with other systems, platforms, humans, and/or products

3) Human system engineering elements, including safety, training, and knowledge, skills, and

abilities required to accomplish functions of the system, and characteristics of information displays and operator controls

4) Characterization of operator(s) and support personnel including special design requirements

and applicable movement, or visual or workload limitations

5) Characterization of information displays and operator controls

6) System characteristics including design limitations (capacity, power, size, weight); technology

limitations (precision, data rates, frequency, language); inherent human limitations (physical

and cognitive workload, perceptual abilities, and reach and anthropometric limitations); and

standardized end items, nondevelopmental items, and reusability requirements

7) Design constraints, including project, enterprise, and external constraints that limit design solutions and/or development procedures

8) Design capabilities and capacities for planned evolutionary growth

6.2 Requirements validation

The project shall perform the tasks of requirements validation. During requirements validation activity

a) The requirements baseline that was established during requirements analysis is evaluated to ensure

that it represents identified stakeholder expectations and project, enterprise, and external constraints.

b) The requirements baseline is assessed to determine whether the full spectrum of possible system

operations and system life cycle support concepts has been adequately addressed.

When voids in needs, constraints, etc., are identified or needs are not properly addressed, requirements

analysis and validation are repeated until a valid requirements baseline is generated. The validated

requirements baseline is documented in the integrated repository and is an input to functional analysis. The

tasks associated with requirements validation are identified in Figure 11.

# Data Model and Description

This section presents various UML diagram to help showcase the functions of this PV wireless monitoring.

## Use Case Diagram

A Use Case Diagram visually captures functions of the system. It shows the interactions between external actors on internal functions of the system. The system boundary helps to define the external actors and the internal functions. The extends signify another use case that is similar, with added functionality. Moreover, use case templates are provided to give detailed information for each use case of the system in the diagram. These templates show which actors are involved for a specific use case and shows the system requirement it satisfies.

TO BE DEFINED

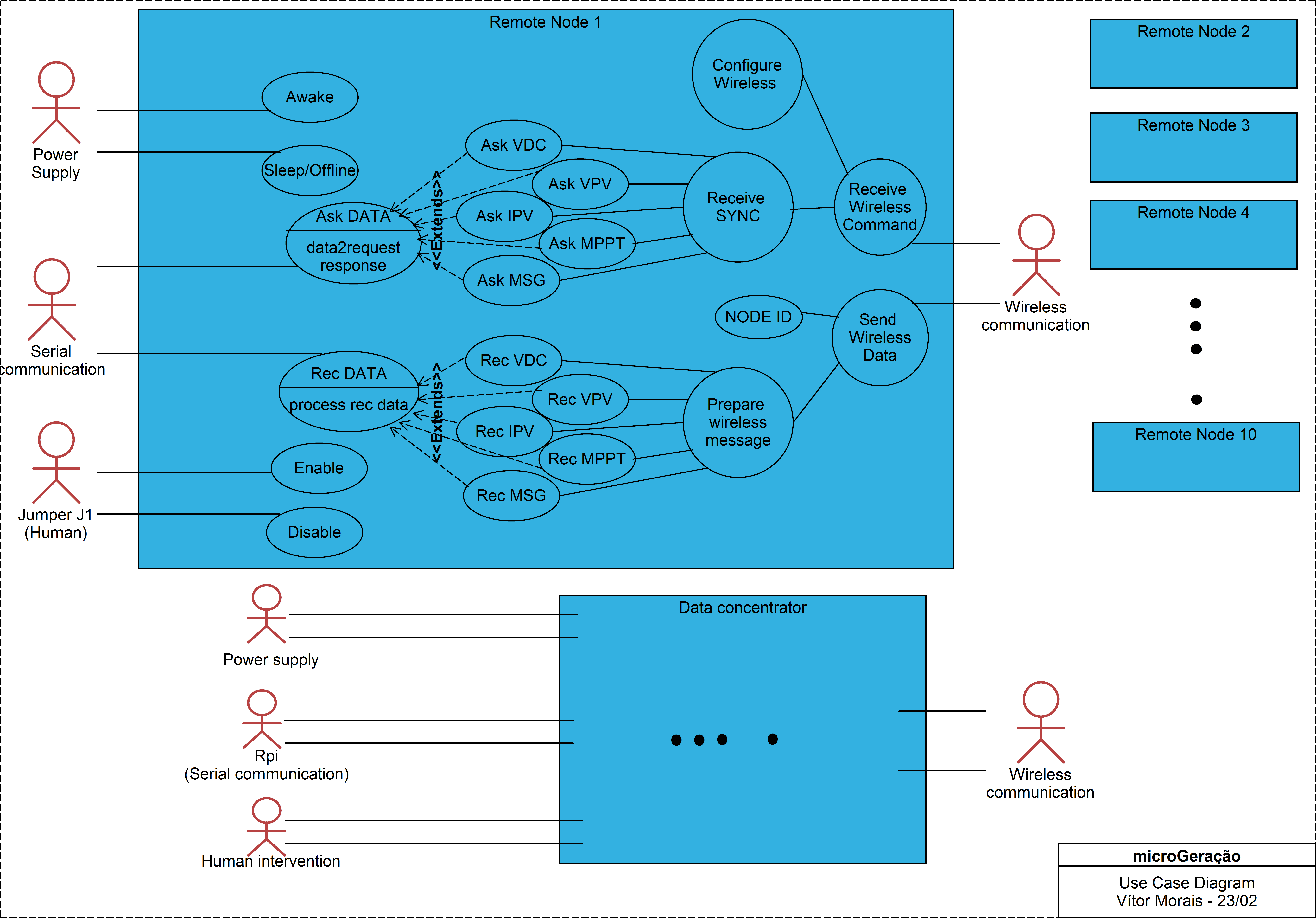


Figure 2 - Use Case Diagram – node

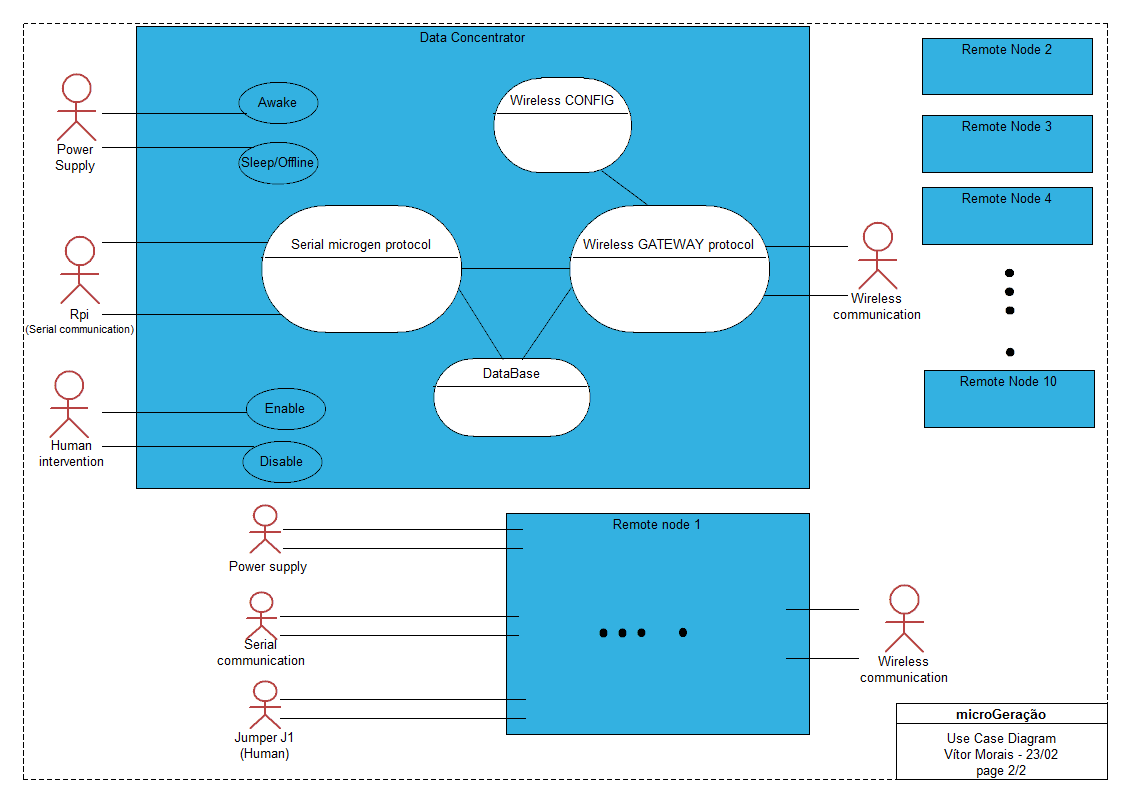


Figure 3 - Use Case Diagram - Data Concentrator

|  |  |
| --- | --- |
| **Use case** | Awake |
| **Actors** | Power Supply |
| **Description** | The system should start up when the system voltage threshold is  above 4.5 volts. |
| **Type** | Primary |
| **Cross-Refs** |  |
| **Use Cases** | All |

|  |  |
| --- | --- |
| **Use case** | Sleep/Offline |
| **Actors** | Power Supply |
| **Description** | The system should go into sleep mode or offline mode when the system voltage threshold is below 3 volts. |
| **Type** | Primary |
| **Cross-Refs** |  |
| **Use Cases** | All but awake |

|  |  |
| --- | --- |
| **Use case** | Enable |
| **Actors** | Human Intervention |
| **Description** | The system should restart and go to enable mode when the user presses a button (or jumper) |
| **Type** | Optional |
| **Cross-Refs** |  |
| **Use Cases** | All but disable |

|  |  |
| --- | --- |
| **Use case** | Disable |
| **Actors** | Human Intervention |
| **Description** | The system should restart and go to disable mode when the user presses a button (or jumper) |
| **Type** | Optional |
| **Cross-Refs** |  |
| **Use Cases** | All but enable |

|  |  |
| --- | --- |
| **Use case** | Serial microgen protocol |
| **Actors** | Rpi (serial communication) |
| **Description** | The system should execute the serial microgen protocol |
| **Type** | Essential |
| **Cross-Refs** |  |
| **Use Cases** | All |

|  |  |
| --- | --- |
| **Use case** | DataBase |
| **Actors** |  |
| **Description** | The system should stores the data and command between serial and wireless |
| **Type** | Essential |
| **Cross-Refs** |  |
| **Use Cases** | All |

|  |  |
| --- | --- |
| **Use case** | Wireless CONFIG |
| **Actors** |  |
| **Description** | The system should have stored all the configuration of remote nodes |
| **Type** | Primary |
| **Cross-Refs** |  |
| **Use Cases** | All but disable |

|  |  |
| --- | --- |
| **Use case** | Wireless GATEWAY protocol |
| **Actors** | Wireless communication (with nodes) |
| **Description** | The system should send a SYNC signal to remote nodes;  The system should receive all data from remote nodes |
| **Type** | Primary |
| **Cross-Refs** |  |
| **Use Cases** | All but disable |

## Domain Model

The following section contains a domain model and a data dictionary for the system.

In the domain model, boxes represent logical objects within the system. Within the boxes, lines preceded with a “­” represent individual pieces of information that that particular object knows, and lines preceded with a “+” indicate actions a given object can perform. Lines connecting the objects show relationships among the boxes, and include a descriptive name and numbers representing how many of each object is involved.

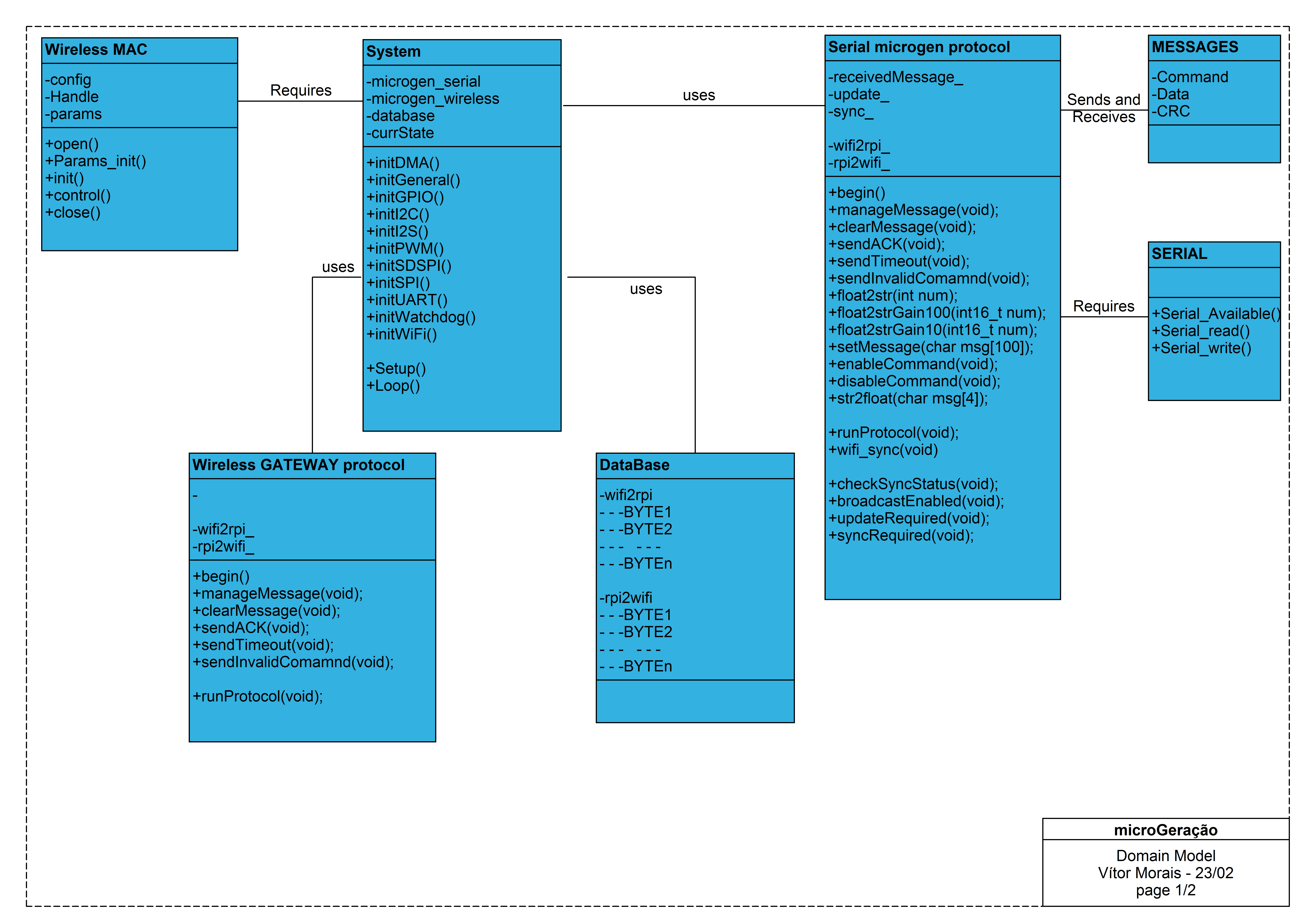


Figure 4 - Domain model diagram – Concentrator

**Data dictionary**

The data dictionary provides additional detail for every element in the domain model.

|  |  |  |
| --- | --- | --- |
| **Element Name** |  | **Description** |
| System |  | Contains the core functionality of the wireless concentrator node such as programming, protocol dependencies, peripheral and communication ports |
| **Attributes** |  |  |
|  | currState | Current state of the system: setup, loop\_waitForSerial, loop\_waitForWifi, loop\_updateDB |
|  | microgen\_serial | Object of serial communication protocol between the data concentrator and the raspberry pi |
|  | microgen\_wireless | Object of wireless communication protocol between data concentrator and wifi nodes |
|  | Database | Object of data storage |
| **Operations** |  |  |
|  | +initDMA() | Initialize board specific DMA settings |
|  | +initGeneral() | This function initializes the general board specific settings. |
|  | +initGPIO() | Initialize board specific GPIO settings |
|  | +initI2C() | Initialize board specific I2C settings |
|  | +initI2S() | Initialize board specific I2S settings |
|  | +initPWM() | Initialize board specific PWM settings |
|  | +initSDSPI() | Initialize board specific SDSPI settings |
|  | +initSPI() | Initialize board specific SPI settings |
|  | +initUART() | Initialize board specific UART settings |
|  | +initWatchdog() | Initialize board specific Watchdog settings |
|  | +initWiFi() | Initialize board specific WiFi settings |
|  |  |  |
|  | +Setup() | General setup of the system |
|  | +Loop() | General loop of the system |

|  |  |  |
| --- | --- | --- |
| **Element Name** |  | **Description** |
| DataBase |  | Contains the data to be exchanged between serial and wireless |
| **Attributes** |  |  |
|  | wifi2rpi : struct | Data exchanged from wifi nodes to rpi |
|  | rpi2wifi : struct | Data exchanged from rpi to wifi nodes |
| **Operations** |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Element Name** |  | **Description** |
| SERIAL |  | Contains the serial Hardware layer (HAL) |
| **Attributes** |  |  |
|  |  |  |
| **Operations** |  |  |
|  | Serial\_Available() | Checks if serial read buffer is available |
|  | Serial\_read() | Reads incoming byte |
|  | Serial\_write() | Writes message to serial port |
| **Element Name** |  | **Description** |
| Serial microgen protocol |  | Contains the class description of the serial protocol implemented in the microgen project |
| **Attributes** |  |  |
|  | currState | Current state of the protocol: setup, run |
|  | receivedMessage: array | Stores temporarily the received message |
|  | update : boolean | Flag: if enabled then updates the database for the new data |
|  | sync : boolean | Flag: if enabled then ask device for new messages |
|  |  |  |
|  | wifi2rpi\_ | Database: stores the data received via wifi to be sent to rpi |
|  | rpi2wifi\_ | Database: stores the data (commands) sent by rpi to the wifi |
| **Attributes** |  |  |
|  | begin() | Initializes the protocol |
|  | +manageMessage(void); | Processes the received message |
|  | +clearMessage(void); | Clears the received message |
|  | +sendACK(void); | Sends via serial a ACK message |
|  | +sendTimeout(void); | Sends via serial a TIMEOUT message |
|  | +sendInvalidComamnd(void); | Sends via serial a INVALID message |
|  | +float2str(int num); | Converts a number to ASCII and sends it via serial |
|  | +float2strGain100(int16\_t num); | Converts a number with 2 decimals to ASCII and sends it via serial |
|  | +float2strGain10(int16\_t num); | Converts a number with 1 decimals to ASCII and sends it via serial |
|  | +setMessage(char msg[100]); | Sets message of receivedMessage array |
|  | +enableCommand(void); | Sets command flag to COMMAND\_ENA |
|  | +disableCommand(void); | Sets command flag to COMMAND\_DIS |
|  | +str2float(char msg[4]); | Converts a received command value from ASCII to float |
|  |  |  |
|  | +runProtocol(void); | Waits and receives serial message |
|  | +wifi\_sync(void) | Enables system for SYNC (or broadcast with remote nodes) |
|  |  |  |
|  | +checkSyncStatus(void); | Disables Sync flag; returns SYNC flag status |
|  | +broadcastEnabled(void); | Broadcasts all received raw data through the serial port |
|  | +updateRequired(void); | Disables UPDATE flag; returns UPDATE flag status |

|  |  |  |
| --- | --- | --- |
| **Element Name** |  | **Description** |
| Messages |  | Data structure of the exchanged serial messages: #DATA#CRC# |
| **Attributes** |  |  |
|  | Data | Data exchanged |
|  | CRC | Number of bytes of DATA |
|  | Command | Optional: it assumes the format {COMMANDVALUE} with 4 ASCII number bytes for COMMAND and 4 ASCII number bytes for data |
| **Operations** |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Element Name** |  | **Description** |
| Wireless GATEWAY protocol |  | Contains the class description of the wireless protocol to be implemented |
| **Attributes** |  |  |
|  | currState | Current state of the protocol: setup, run |
|  |  |  |
|  | wifi2rpi\_ | Database: stores the data received via wifi to be sent to rpi |
|  | rpi2wifi\_ | Database: stores the data (commands) sent by rpi to the wifi |
| **Attributes** |  |  |
|  | begin() | Initializes the protocol |
|  | +manageMessage(void); | Processes the received wifi message |
|  | +clearMessage(void); | Clears the received message |
|  | +sendACK(void); | Sends via wifi to a node a ACK message |
|  | +sendTimeout(void); | Sends via wifi to a node a TIMEOUT message |
|  | +sendInvalidComamnd(void); | Sends via wifi to a node a INVALID message |
|  |  |  |
|  | +runProtocol(void); | Waits and receives serial message |

|  |  |  |
| --- | --- | --- |
| **Element Name** |  | **Description** |
| Wireless MAC |  | Contains the wifi Hardware layer (HAL) |
| **Attributes** |  |  |
|  | config | Configuration parameters |
|  | Handle | Object handler |
|  | Params | Object parameters |
| **Attributes** |  |  |
|  | +open() | Function to initialize a given WiFi peripheral |
|  | +Params\_init() | Function to initialize the WiFi\_Params structure to its defaults |
|  | +init() | Function to initialize the WiFi module |
|  | +control() | Function performs implementation specific features on a given WiFi\_Handle. |
|  | +close() | Function to close a WiFi peripheral specified by the WiFi handle |

**Remote node vs data concentrator**

The code implemented in each remote node should be similar to the data concentrator. On the serial communication, the microgen protocol should be implemented in both the PV converter and the remote node. On the wifi communication, the remote node must implement a Medium Access Control similar based on a producer-consumer mesh network.

## Sequence Diagrams

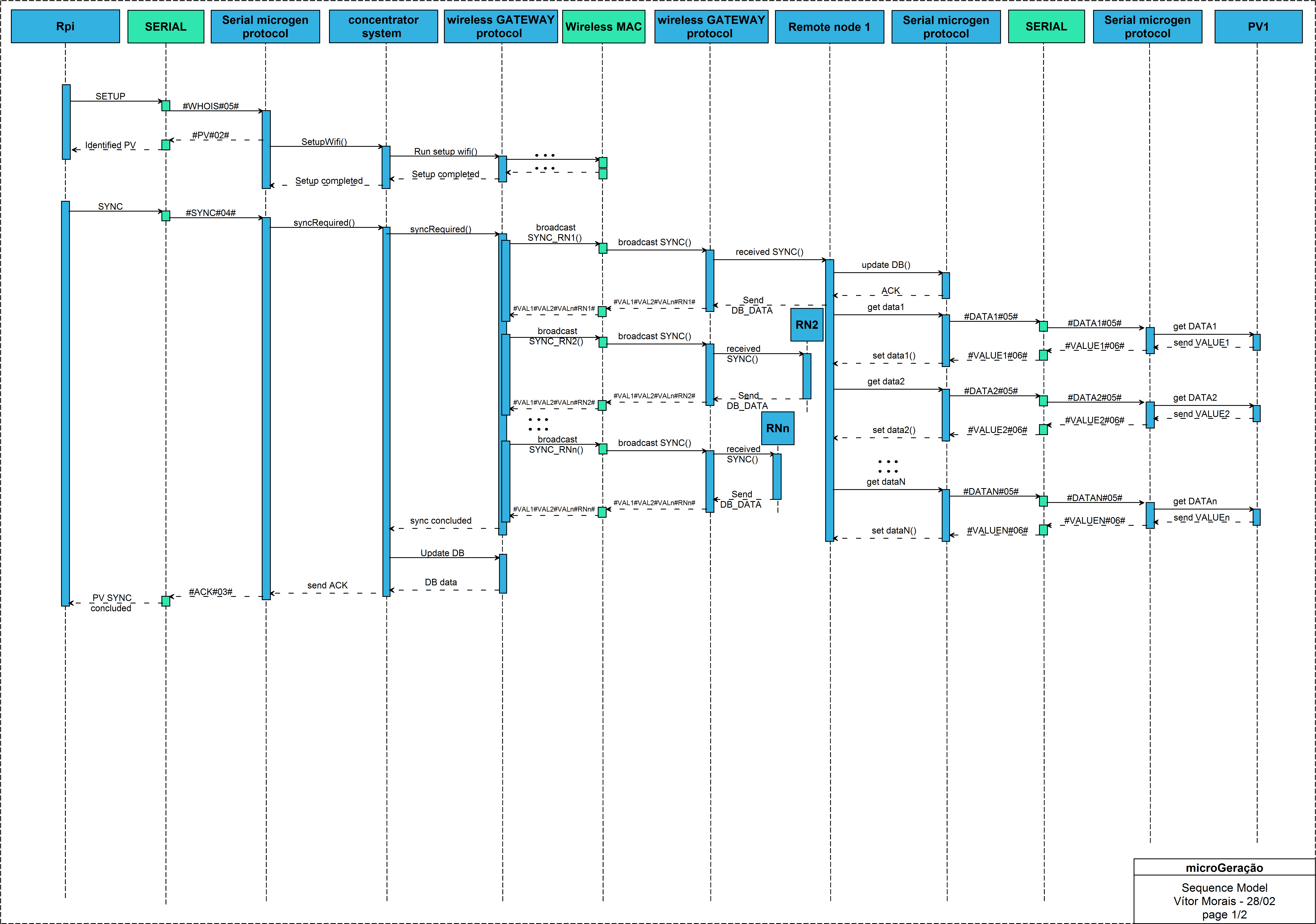


Figure 5 - Sequence diagram for SETUP and SYNC.

Figure 5 is intended to present the SETUP and SYNC sequences. Setup is a Raspberry routine intended to identify the devices connected itself. Sync routine provides a signal to all devices in order to hold the current variables in memory (dataBase). The signal SYNC must be broadcasted to devices at the same time, with a **maximum jitter of 10ms**.

The routine in the data concentrator device collects all data from remote nodes after a SYNC signal. The data collection from remote nodes should include the serial microgen protocol in both the remote node and the PV power converter.

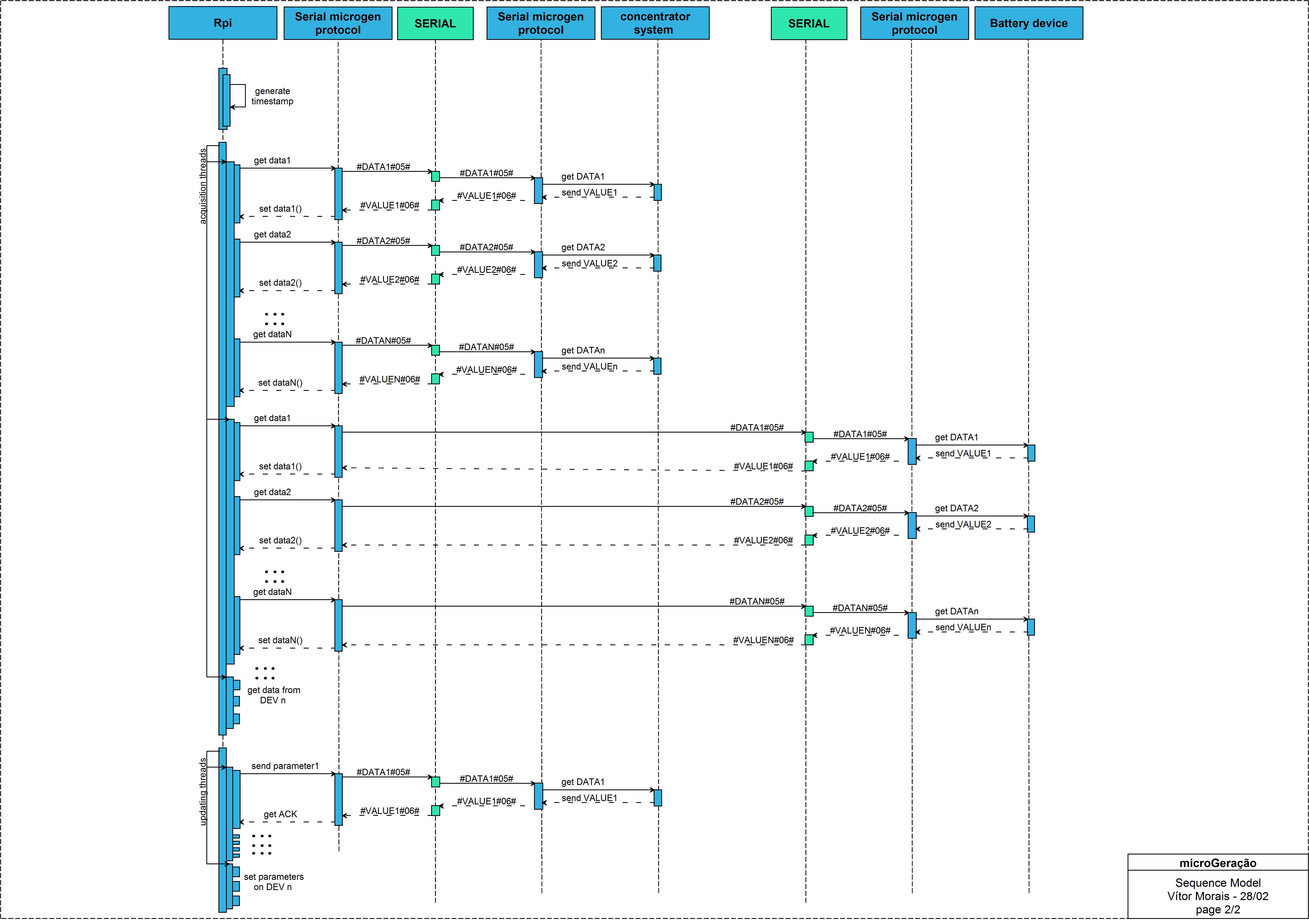


Figure 6 - Sequence diagram for timestamp, acquisition and updating.

After the last SYNC signal acknowledge received, the master should hold the current timestamp.

The acquisition of the data from each device should be processed simultaneously. From the point of view of the PV data concentrator device, all data should be transmitted using the serial microgen protocol.

The commands and references must be sent to devices after the acquisition. The same protocol should be used to send the data from the master to the data concentrator of the PV power converters.

## State Diagram

The figure 7 shows all possible states and transitions for the PV wireless monitoring. States are marked using a round­edged box with a short description of the state. State transitions are signified using arrows with text descriptions. These descriptions generally consist of a function call and any constraints that need to be met before the transition is made. Constraints are indicated using brackets. The starting state of a diagram is indicated using a black dot pointing to a state.

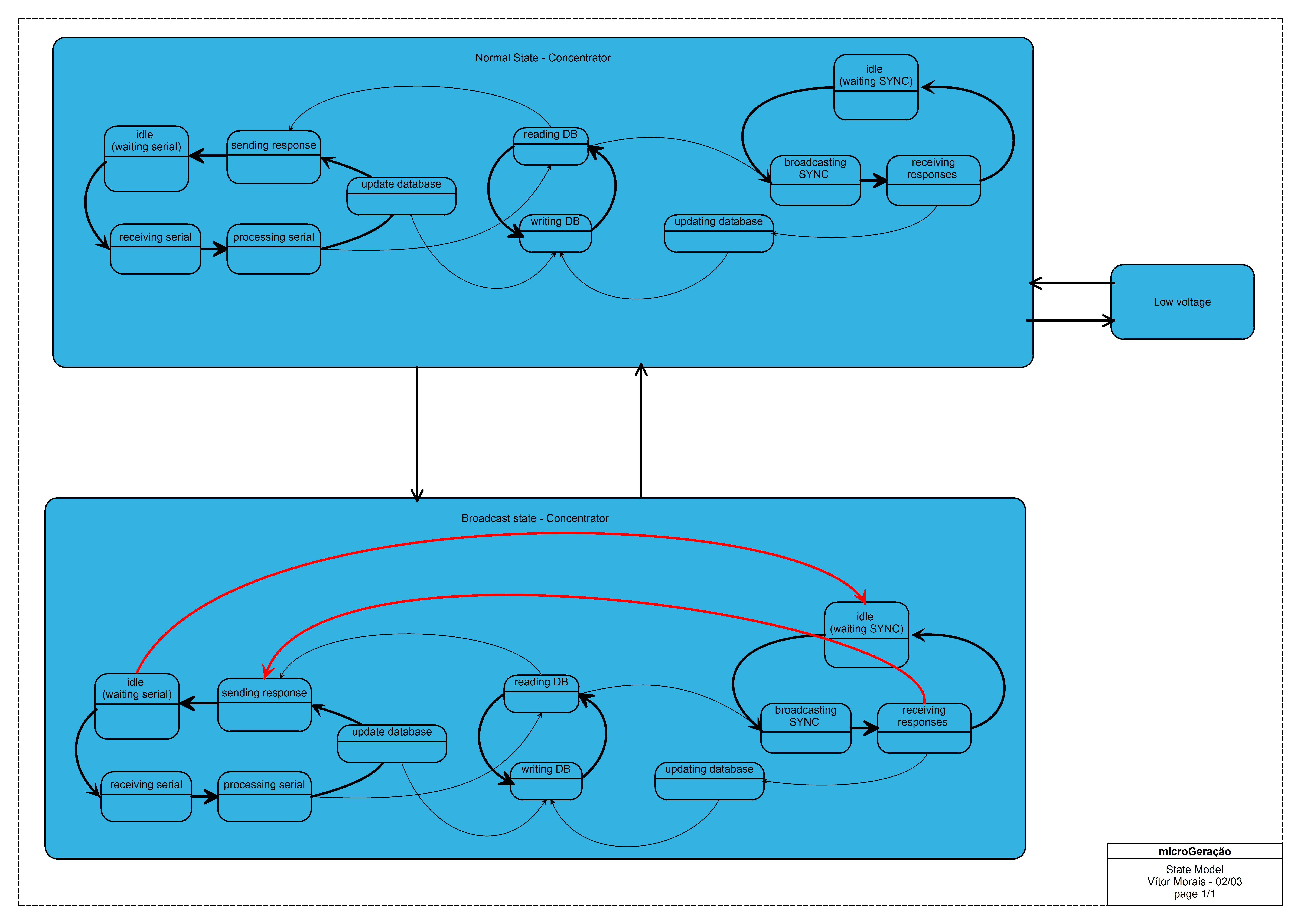


Figure 7 - State diagram for the PV wireless monitoring subsystem.

## Prototype

The prototype visually simulates what happens on the system in different scenarios.

### Prototype Operation

The main result of the PV wireless monitoring subsystem is accessible with a browser through the raspberry pi master interface.

A alternative result will be the access to the remote node using the serial port through the broadcast operation mode

### Sample Scenario

DATA TRANSFER

REMAINING SUPPORTED SCENARIOS:

# References