Implementing an Intermittent Computing System for an Autonomous Ocean Analyzer Prototype

Documentation

Advisor: Dr. Haonan Wang

Team Members:
Milan Dudhatra
Virag Bhanderi
Moxank Patel

Conceptual Design

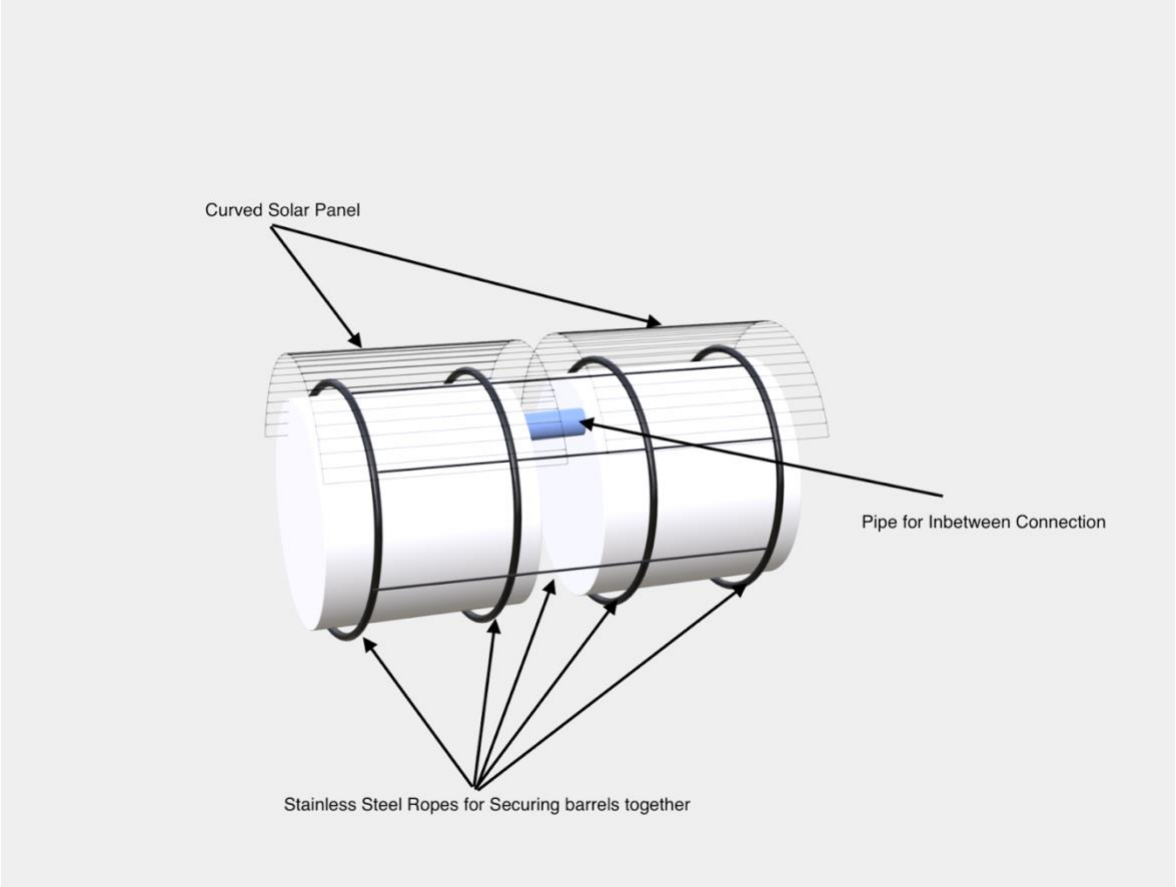


Fig 1. Conceptual Design of the System

The design illustrated in Fig 1. exhibits a sophisticated integration of safety and modularity. Central to its construction is incorporating stainless steel ropes, meticulously designed to interconnect and fortify the barrels. These ropes ensure structural integrity and provide a reliable means of assembly. Moreover, the curved solar panel, strategically positioned atop, underscores a commitment to sustainable energy utilization.

The embodiment of modularity is evident in the intricate internal configuration. Each component is housed within individualized units crafted from durable PVC sheets. This modular approach facilitates effortless addition or removal of components, enabling seamless maintenance and adaptability to evolving requirements.

Safety remains paramount in this design. A specialized layer of foam envelops the internal housing units. This isn't just any foam; it possesses fire-resistant properties, effectively mitigating fire hazards and ensuring the utmost protection of the internal contents, which may include sensitive materials like batteries and reagents.

Furthermore, the blue pipe, tailored for interconnection, exemplifies a forward-thinking approach to system integration. Its design prioritizes waterproofing, a critical feature considering potential exposure to aqueous environments.

In conclusion, this design presents an avant-garde amalgamation of safety and modularity. Its forward-thinking features, derived from a meticulous analysis of Fig1, make it an attractive proposition for those seeking reliability, adaptability, and utmost safety in their systems.

Replicating the work

Hardware: Above table components
Software: GitHub Repo https://github.com/Moxi3231/pFiona-CMPE-295
Clone the above repository

Repository Structure and Contents

The GitHub repository for this project serves as a central hub for all the relevant code and documentation. The structure of the repository is designed for ease of navigation and understanding.

User Interface

- **Location**: Artifacts/User Interface
- **Description**: The user interface of the project is a web application based on the Django framework. It provides a comprehensive and interactive platform for managing and monitoring the system's functions.
- **Execution Instructions**: To run the user interface, navigate to the main directory of the User Interface folder and execute the command python3 manage.py runserver. This will launch the Django server, allowing access to the user interface.

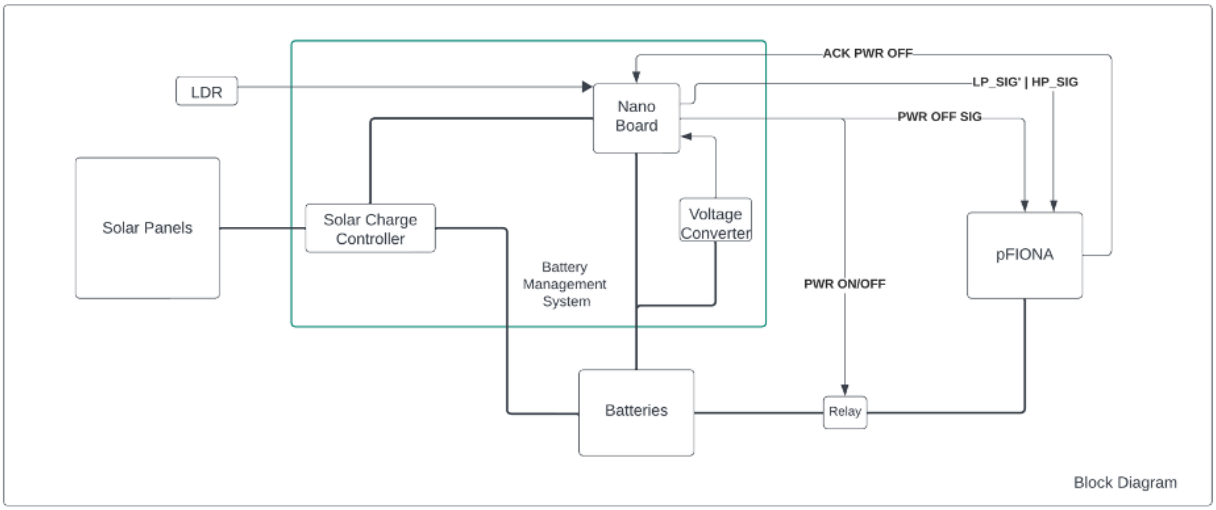
Power Management System Code

- **Location**: Artifacts/BMS/main
- **Description**: This folder contains the Arduino code for the power management system. This code is a crucial component of the project as it governs the energy management, ensuring efficient and sustainable operation of pFiona.
- **Deployment Instructions**: The code needs to be flashed onto the memory of an Arduino board using the Arduino IDE. This process is essential for integrating the power management system with the physical hardware.

Miscellaneous Files

- **Location**: misc folder
- **Contents**: This folder includes various supplementary materials that support the project. It encompasses the design aspects of the system, detailed reports, and other relevant files that provide additional context and information about the project.

Procedure for Setting Up a Solar-Powered System with Voltage Stabilization and Battery Integration



I. Introduction

This guide outlines a comprehensive procedure for setting up a solar-powered system with an emphasis on voltage stabilization and battery integration. The system utilizes a 24V power supply, Lifepo4 batteries, a solar charge controller, solar panels, a voltage stabilizer, and a voltage converter. This setup is designed to power a nano board and a Raspberry Pi effectively.

II. Equipment and Materials Required

Group 1: Items needed to verify the system design

#	Item	Additional Comments	Quantity	Check Mark
1	Wires (24 Guage)		1	C
2	Wires (20 Guage)		1	C
3	Light Sensor		2	
4	Batteries (15 AH)	12 Volts Each	2	C
5	Relay Switch		4	C
7	Voltage Converter (5V)		2	C
8	Voltage Converter (24V)		1	C
9	Clock Module	For Nano Board	1	C
10	Cells for Clock Module	Pack of 12	1	C
11	Arduino Nano		1	X C

#	Item	Additional Comments	Quantity	Check Mark
1	Wires (24 Guage)		1	C
2	Wires (20 Guage)		1	C
3	Light Sensor		2	
4	Batteries (15 AH)	12 Volts Each	2	C
5	Relay Switch		4	C
7	Voltage Converter (5V)		2	C
8	Voltage Converter (24V)		1	C
12	Wires Pair	For connecting solar Panel in series	1	
13	Extension Wire	For extending the connection of Panel	2	
14	Solar Charge Controller MPPT		1	
15	Solar Charge Controller PWM		1	
16	5V Converter	For powering NANO	2	
18	Wires	12 Guage	2	
19	Converter			

Note: Check Mark X means already owned. Check Mark C means package received.

1. **Wires (24 Gauge)**: Connect Nano Board, Light Sensor, and other parts.
2. **Wires (20 Gauge)**: Thicker wires are used where more current is required (Battery to pFIONA Controller Board).
3. **Light Sensor**: Detects ambient light, useful in optimizing solar panel efficiency.
4. **Battery 15 AH**: Stores electrical energy essential for the system's power supply (12 V each). To be connected in parallel to get 24 V DC.
5. **Relay Switch**: Controls the electrical circuit, allowing or stopping the current flow.
6. **Solar Panel**: Consists of 2 panels and a kit that acts as the main source of power generation.
    a. Charge Controller: It controls the charging and prevents the battery from being overcharged and drained completely by cutting the supply from the solar panel.
    b. The kit consists of a charge controller and connecting wires for solar panel
7. **Voltage Converter 5V**: Monitors the voltage in the system, which is crucial for maintaining appropriate energy levels. Nano Board is compatible with a voltage range of 0-5V, so to detect the battery level, this converter is used.
8. **Voltage Regulator 24V**: Ensures the voltage supplied to components is consistent and within safe limits. Battery at full charge can deliver more than 24 V and may damage the controller board of pFIONA. To prevent that, this regulator is essential, and this will also act as a stabilizer.
9. **Clock Module**: Essential for time-keeping functions, specifically for the Arduino Nano.
10. **Cells for Clock**: Powers the clock, ensuring continuous operation.
11. **Arduino Nano**: Acts as the brain of the project, processing inputs and controlling various components.

Group 2: items needed for deployment in seawater:

#	Item	Additional Comment	Quantity	Check Mark
1	Silicone Sealant for Wires		2	
2	Water Phobic Film		1	
3	Rope	Stainless Steel Rope	1	
4	Foam	Pack of 3	1	
5	Foam (Fire Proof)	Set of 3 + Gun + Foam Cleaner	1	
6	Barrels		2	
7	Water Pump		4	
8	Pipes for water Pump		4	
9	Anchor for Barrels	Also contains ropes	1	
10	Superglue		2	
11	Wrap Tubes	2 inch	1	
12	Wrap Tubes 1 Inch	1 inch	1	
13	2 inch Pipe	Acrylic Tube	1	
14	PVC Sheets		3	
15	Hot Glue Gun with Sticks		1	
16	Heat Gun		1	
17	PVC GLUE		2	
18	Cement	80 Lb Bags	1	
19	Electrical Gloves		3	X
20	Solder		1	X
21	Wire Cutter		1	X
22	Soldering Machine		1	X
23	Hooks	10 Pieces	1	
24	String Saw		1	
25	PVC Tubes	2 Inch Diameter	1	
26	Sand Paper		1	
27	Inlet Balls	Pack of 15	1	
28	Screw Driver	Set of Screw Driver	1	
29	Screws		1	
30	Pipe filter	Pack of 100	1	
31	Relay Switch		2	

1. **Silicone Sealant for Wires**: Ensures that wire connections are insulated and protected from environmental factors.
2. **Water-Phobic Film**: Applied to components for protection against water and moisture.
3. **Rope**: Used for securing components in place. It will be mainly used to secure the barrels and strengthen wiring.
4. **Foam**: Used for padding protecting sensitive components.
5. **Foam (Fire Proof)**: Includes a set of three, a gun, and foam cleaner, offering additional protection against fire.
6. **Barrels**: Utilized for storage purposes within the system. The barrels will be filled with cement that will act as a weight for them and allow barrels to be submerged in water. See below for the calculation of weight.
7. **Water Pump**: Aspirate water from the ocean and store it in a container.
8. **Pipes for Water Pump**: Necessary for directing water flow for the pump above Water Pump (Item 7).

9. **Anchor for Barrels**: Provides stability and secure barrel positioning. Rope and anchor are available in the set.
10. **Superglue**: Essential for bonding and securing various components.
11. **Wrap Tubes**: Used for waterproofing wires and wrapping them together.
12. **Wrap Tube 1 inch**: Used for waterproofing wires and wrapping them together.
13. **2-inch Pipe**: Facilitates connections between both barrels.
14. **PVC Sheets**: Useful for casing and protecting various system parts.
15. **Hot Glue Gun**: A tool used for applying hot glue, useful for securing and insulating connections. (Note: "with Sticks" is not included in the original list; the item here is just the hot glue gun.)
16. **Heat Gun**: For shrinking wrap tubes and bending plastics.
17. **PVC GLue:** For connecting PVC boards and pipes.
18. **Cement**: For filling the barrels. The Mix Ratio is usually around 0.4 to 0.7 (Water/Cement), the lesser it is, the more strength it will have. Considering the other materials that will be added to cement besides water for strong bonding. The cement will constitute around 40-50 percent, the water will be around 20-25 percent, and the remaining will be sand and stones/gavels or other materials.
19. **Electrical Gloves**: Ensures safety by protecting against potential electrical shocks. (Already Available)
20. **Solder**: Used with the soldering machine for connecting components. (Already Available)
21. **Wire Cutter:** Cutting Wires.
22. **Soldering Machine:** Soldering the connection. (Already Available)
23. **Hooks:** To hang the reagents bag.
24. **String Saw:** Cutting the PVC Sheet and Pipe.
25. **PVC Tube:** For connecting the barrels (hides electrical wiring).
26. **Sand Paper:** For smoothing the connection.
27. **Inlet Balls:** Will allow inlet pipe to stay afloat
28. **Screw Driver:** Used for the screws.
29. **Screws:** Securing the Nano Board, Light Sensor, and other components
30. **Filter:** Filtering the water from the inlet.
31. **Relay Switch**: Control the inlet and outlet pump..

III. Procedure

A. Establishing the Power Supply

1. **Battery Configuration for 24V Supply**
    - Begin by setting up a stable 24V power supply.
    - Connect two 12V batteries in series.
    - Ensure correct polarity and secure connections to form a 24V system.

B. Integration with Solar Charge Controller

1. **Series Connection to Charge Controller**
    - With the batteries in series, proceed to connect them to the solar charge controller.
    - Follow the manufacturer's guidelines meticulously.
    - Avoid connecting the solar panel directly to the charge controller without the battery intermediary to prevent damage.
2. **Manual Reference**
    - Consult the charge controller's manual for precise instructions.
    - Adhere to these guidelines to avoid installation errors.

C. Solar Panel Integration

1. **Connecting Solar Panels**
    - Connect the solar panels in series.
    - Then, link them to the charge controller.
2. **Verification**
    - Check the charge controller’s display for proper connectivity and power flow indicators.

D. Configuring the Charge Controller

1. **Controller Setup for Lifepo4 Batteries**
    - Configure the charge controller for Lifepo4 battery compatibility.
    - Refer to the manual for detailed configuration instructions.

E. Voltage Stabilizer and Converter Setup

1. **5-Volt Output Connection**
    - Connect a 5-volt output directly to the battery for the nano board.
    - Establish another 5-volt connection for the Raspberry Pi.
2. **Voltage Stabilizer Connection**
    - Link the voltage stabilizer to the battery to ensure consistent voltage supply.
3. **Voltage Converter Configuration**
    - Connect the voltage converter between the battery and the Arduino’s analog pin A1.

- This step adapts the voltage levels suitable for the Arduino board.

F. Sensor and Grounding

1. **LDR Sensor Integration**
    - Connect the LDR sensor to the load (+5V) and ground pins.
2. **Pull-down Resistor Setup**
    - Attach a pull-down resistor to the ground pin.
    - Connect the LDR sensor's output to the nano board's digital pin A3.

G. Firmware Flashing

1. **Finalizing the Setup**
    - Once all connections are established, proceed to flash the firmware onto the nano board.
    - Utilize the code provided in the repository.
    - Follow the repository's detailed instructions for the flashing process.

IV. Conclusion

Upon completion of these steps, your solar-powered system with voltage stabilization and battery setup should be operational. Regularly check connections and monitor system performance to ensure optimal functioning.

Comprehensive Guide to Setting Up and Operating the pFiona System

I. Introduction

This document provides a detailed, step-by-step guide for setting up and operating the pFiona System, a Raspberry Pi-based sample processing system. It encompasses procedures for initiating scripts on the Pi, installing a Django application, and managing pFiona system operations.

II. Preparing the System

A. Initiating Scripts on Raspberry Pi

1. **Running the Initiate_State_Cycle.py File**
    - Navigate to the pFiona_run_cycle folder.
    - Execute the script using the command:
    - python Initiate_State_Cycle.py
2. **Executing the command_server.py File**
    - Switch to the Server folder.
    - Run the file with the command:
    - python command_server.py

III. Setting Up a Django Application

A. Prerequisites

1. **Python Installation**
    - Confirm that Python is installed on your system.
    - If not, download it from the official Python website.
2. **Virtual Environment Setup**
    - Install virtualenv using:
    - pip install virtualenv
    - Create a virtual environment named pFionaUIenv (already available in the folder).
    - Activate the environment:
        - For Windows:
        - .\pFionaUIenv\Scripts\activate
        - For Unix or MacOS:
        - source pFionaUIenv/bin/activate
3. **Installation of Required Packages**
    - Install necessary Python packages using:
    - pip install -r requirements.txt

(Note: The requirements file is provided in the code directory and the packages might already be installed.)

B. Django Project Configuration

1. **Project Setup**
    - Create a new Django project (if not already created) with:
    - django-admin startproject pFionaUI
    - Navigate to the project directory:
    - cd pFionaUI
2. **Running the Development Server**
    - Start the Django development server with:
    - python manage.py runserver
    - Access the Django application via a web browser at http://localhost:8000/ or http://127.0.0.1:8000/.

IV. Operating the pFiona System

A. System Overview

1. **Components of pFiona System**
    - The pFiona system integrates various components like a centralized dashboard, logging features, data visualization tools, and PI control mechanisms.

B. Dashboard Utilization

1. **Accessing Real-time Insights**
    - Utilize the dashboard for a comprehensive view of the system's status and device information.
2. **Logs Management**
    - Monitor system events through Debug Logs, Operational Logs, and All Logs for efficient troubleshooting.
3. **Data Visualization**
    - Engage with user-friendly tables and dynamic charts for an in-depth analysis of the system's performance.
4. **Graphical Representations**
    - Analyze data trends through dynamic charts, including wavelength versus samples representation.

C. PI Control and Script Operation Modes

1. **Script Execution Modes**

    - Select from Automatic, Pause, Full Manual, User-specified, and Command Line Modes for script operation, based on requirements.
2. **Real-time System Adjustments**
    - Optimize operations using the State Machine for precise control.

D. Establishing Connection with PI

1. **Initiating and Terminating Connection**
    - Start the connection by pressing "Start Connection with PI" and wait for 10-20 seconds.
    - Terminate the connection by selecting "Stop Connection with PI."
2. **Resetting Connection**
    - Use "Enable Reset Connection" followed by "Reset Connection" for troubleshooting.
3. **Error Management**
    - Address connection issues with the "Enable Reset Connection" feature.

V. Conclusion

This guide presents a structured approach to setting up and managing the pFiona System. Adherence to these procedures will ensure a smooth installation and operation of the system. For additional details or further clarification on any section, please feel free to request more information.