Solar Charging Station for Micro-Mobility and Portable Devices User Manual

Supervising Faculty:

- Dr. M. A. Rafe Biswas
- Dr. Reem Salim

Team Members:

- Ahmad Alhjouj
- Ahmed Ismail
- Hector Villasana
- Marvin Villagra
- Maryam Mehryoon
- Mustafa Kerem Yucedag

The University of Texas at Tyler
Department of Mechanical Engineering
College of Engineering - Houston Engineering Center

Table of Contents

Table of Figures

Introduction:

This manual is prepared to guide the users through the process of understanding, setting up, and safely operating the charging station. The system is designed to harness renewable energy, primarily from solar power, supplemented by hydrogen fuel cells, to provide a reliable and sustainable solution for charging micro-mobility devices and portable electronics, even in the absence of sunlight. Additionally, the station integrates an IoT-based data acquisition (DAQ) **system** that enables real-time monitoring, data logging, and performance analysis, enhancing both usability and educational value.

Safety Precautions for Installation:

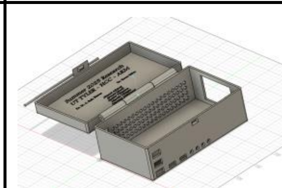
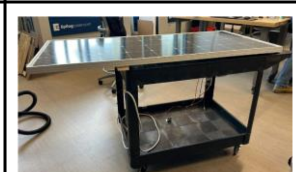
1. Work in a Dry Environment: Always ensure that the installation area is dry. Water is a good conductor of electricity and can lead to electric shocks.

2. Use Insulated Tools: Always use insulated tools when working with electrical components to prevent electric shocks.
3. Wear Personal Protective Equipment (PPE): Wear safety glasses, gloves, and sturdy footwear during the installation process to protect yourself from potential injuries.
4. Avoid Working Alone: It’s safer to work with a partner who can assist you and call for help in case of an emergency.
5. Follow Manufacturer’s Instructions: Always follow the manufacturer’s instructions when installing solar panels, batteries, inverters, and other components. Incorrect installation can lead to equipment damage and safety risks.
6. Proper Handling of Batteries: Batteries can be heavy and contain harmful chemicals. Always handle them with care, use proper lifting techniques, and avoid short-circuiting the terminals.
7. Proper Wiring: Ensure all the wiring is done according to the specified guidelines. Incorrect wiring can lead to fires or equipment damage.
8. Avoid Direct Sunlight Exposure: When installing solar panels, try to avoid direct sunlight exposure to prevent overheating and potential burns.
9. Regular Maintenance: Regularly inspect and maintain all components of the solar charging station to ensure they are in good working condition and to prevent any potential hazards

System Components:

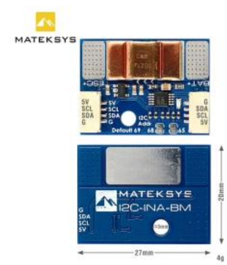
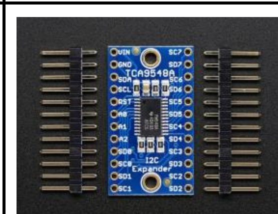
Charging Station Components:

Component	Description	Image
Solar Panel	This 195W monocrystalline panel converts sunlight into DC power, ideal for 12V systems. It outputs 18V at 10.83A (Vmp/Imp) with a Voc of 21.6V. Compact and efficient, it's a reliable source for off-grid charging when paired with an MPPT controller.	

Batteries	A 12.8V, 20Ah (256Wh) lithium battery with a built-in BMS for protection. Offers 3000+ cycles, high energy density, and safe, stable storage for solar energy. Ideal for lightweight, long-life applications.	
Invertor	Converts 12V DC from the battery to 120V AC pure sine wave. Delivers 2000W continuous / 4000W surge power. Safe for electronics, includes USB ports and protections, ideal for powering home appliances off-grid.	
Charge Controller	An MPPT controller that optimizes solar input to charge 12V/24V batteries. Supports up to 30A current and 100V PV input. Features include an LCD display, smart protections, and compatibility with lithium batteries.	
DAQ System Box	3D-modeled and printed to house sensors, and related electronic components used in the data acquisition system. It includes dedicated ventilation holes for thermal regulation, side cutouts for connections between the sensors and cables routing.	
Cart Structure	The solar charging station is built on a two-level wheeled cart for portability. The lower level houses key components such as batteries, inverter, fuel cell, hydrogen generator, the upper-level houses the charge controllers, and the DAQ system. The solar panel is laid flat on top for easy transport and storage.	

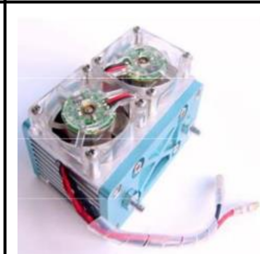
DAQ System Components:

Component	Description	Image

Arduino (Mega-2560 R3)	The Arduino Mega 2560 R3 is a microcontroller board based on the ATmega2560 chip. It is designed for more complex projects that require a large number of I/O pins, memory, and serial interfaces. It is ideal for advanced robotics, 3D printing (e.g., Marlin firmware), and IoT systems.	
STM32(B-L475E-IOT01A2)	The STM32 B-L475E-IOT01A2 is an IoT-focused development board featuring an STM32L475VG ultra-low-power ARM Cortex-M4 microcontroller. It integrates multiple sensors, connectivity modules, and expansion options, making it ideal for IoT prototyping, cloud connectivity, and edge computing.	
MATEKSYS(I2C-INA-BM)	The Matek I2C-INA-BM is a current and voltage sensor module designed for flight controllers and embedded systems. It uses the Texas Instruments INA226 chip for high-side current sensing over I2C and includes a shunt resistor and a Bosch BMP280 barometric pressure sensor.	
Thermocouples (MCP9601)	The MCP9601 is a highly accurate thermocouple amplifier and digital converter developed by Microchip Technology. It supports direct interface with standard thermocouples (types K, J, T, N, S, E, B, and R) and converts the analog thermoelectric voltage to a temperature reading via an I²C interface	
Multiplexer (TCA9548A)	The MCP9601 is a highly accurate thermocouple amplifier and digital converter developed by Microchip Technology. It supports direct interface with standard thermocouples (types K, J, T, N, S, E, B, and R) and converts the analog thermoelectric voltage to a temperature reading via an I²C interface	
TFT LCD Display 3.5 inch	A 3.5-inch TFT LCD Display is a compact, full-color screen based on Thin-Film Transistor (TFT) technology. It typically supports touch input (resistive or capacitive), features a 320×480 pixel resolution, and connects via SPI or parallel (8/16-bit) interfaces depending on the model.	

XT60E-F Cable	The XT60E-F is a panel-mount female XT60 connector with cable leads, designed for secure, high-current DC connections in battery-powered and embedded systems. The "E" denotes the panel-mountable (embedded) version, while the "F" denotes the female socket.	
K Type Panel Mount Thermo	A K-Type Panel Mount Thermocouple Connector is a fixed installation interface for K-type thermocouples, allowing easy connection and disconnection of thermocouple wires on an instrument panel, enclosure, or DIN box. It provides a reliable, high-temperature, and low-noise electrical connection for accurate thermocouple signal transmission.	
jst-gh 4pin quick connector		

Hydrogen Fuel Cells components:

Component	Description	Image
Stack	It includes a plurality of plate-like fuel cells arranged along an axis generally parallel to cell thickness with electrically conductive separator plates between each pair of cells.	
Controller	It controls the purging and short circuiting of the stack.	

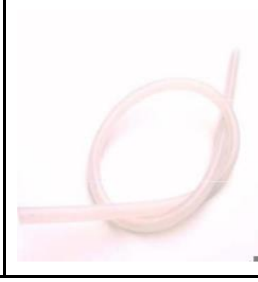
Purge valve	It controls the purging time for purging water and redundant Hydrogen from the fuel cells.	
Tube	It is used for connecting the pressure regulator to the stack, the stack to the purge valve and the purge valve to a place away from the stack.	

Figure 1: Cart structure

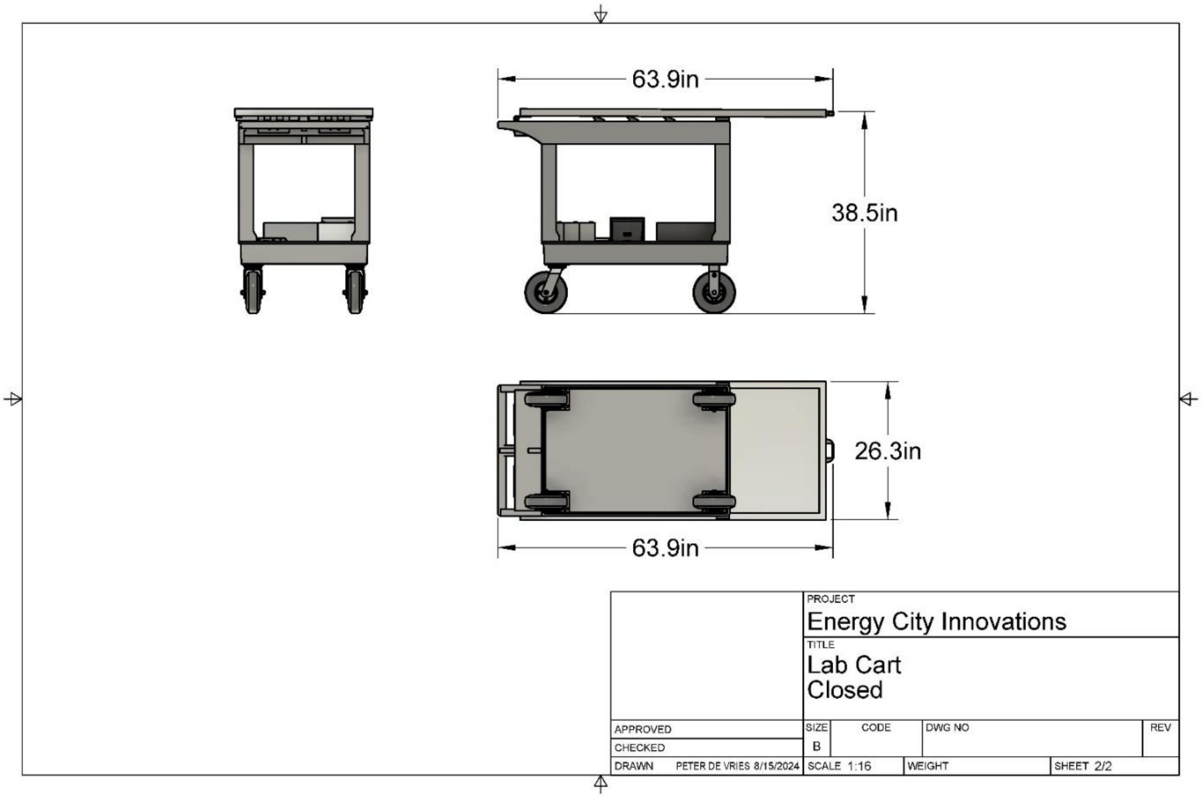


Figure 2: Lab Cart in closed position

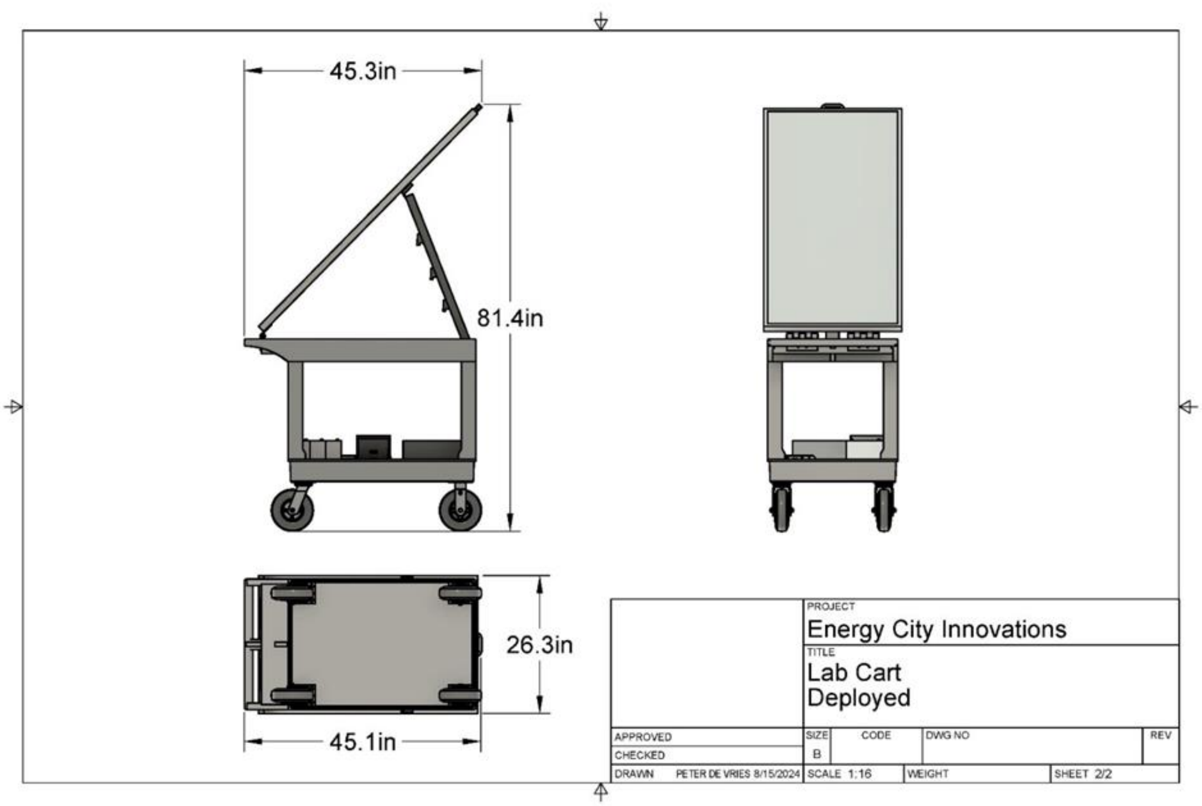
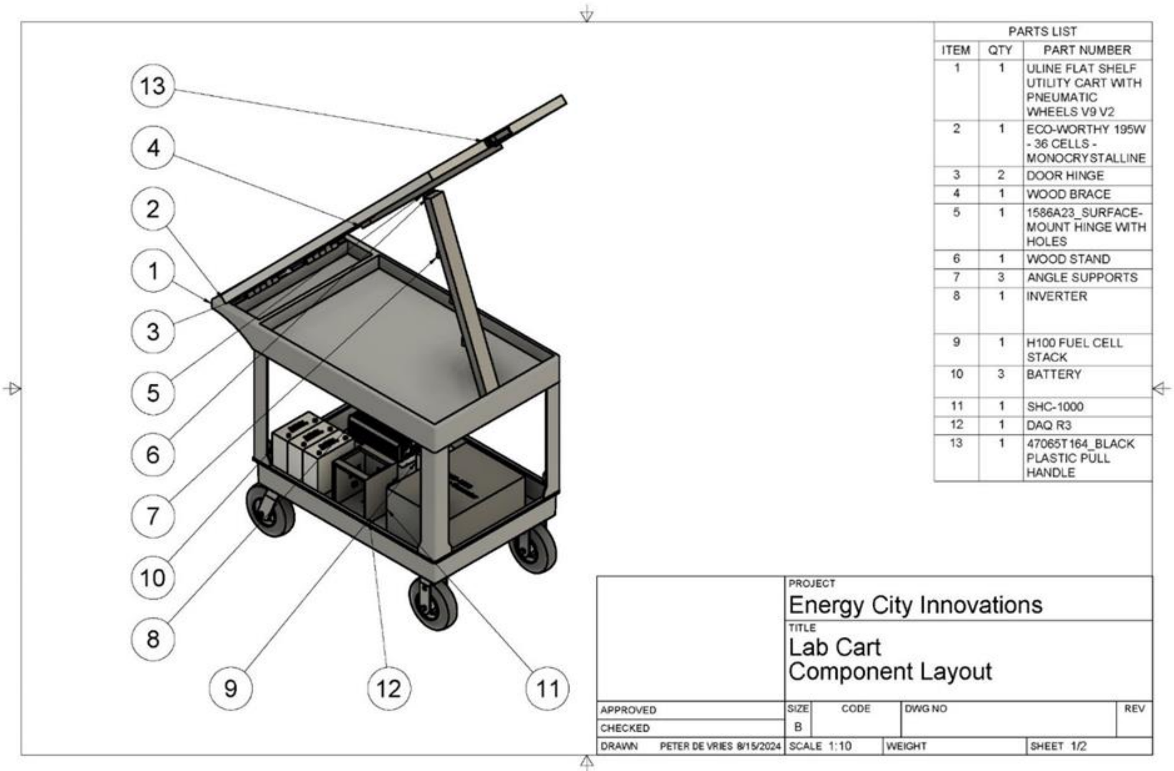


Figure 3: Lab Cart in open position



PARTS LIST

ITEM	QTY	PART NUMBER
1	1	ULINE FLAT SHELF UTILITY CART WITH PNEUMATIC WHEELS V9 V2
2	1	ECO-WORTHY 195W - 36 CELLS - MONOCRYSTALLINE
3	2	DOOR HINGE
4	1	WOOD BRACE
5	1	1586A23_SURFACE-MOUNT HINGE WITH HOLES
6	1	WOOD STAND
7	3	ANGLE SUPPORTS
8	1	INVERTER
9	1	H100 FUEL CELL STACK
10	3	BATTERY
11	1	SHC-1000
12	1	DAQ R3
13	1	47065T164_BLACK PLASTIC PULL HANDLE

Figure 4: Callouts for all the parts and materials

System Design & Assembly:

System Layout:

The solar charging station is mounted on a two-tier wheeled cart for easy portability. used for housing additional components such as batteries, inverter, fuel cell, hydrogen generator while the upper tier holds the charge controllers and data acquisition (DAQ) system. A solar panel is positioned flat on top of the cart for convenient transport and storage, the layer has been designed this way so it can house the whole system and it's wiring efficiently. The panel is placed on a rugged plastic wheel cart, making it portable and easy to move to different locations as needed. This feature enhances the flexibility and utility of the solar charging station as shown in the figures below. The following figures show the general layout for the two levels for the cart. The solar panel is laid flat on top of the cart. This position is for transportation and storage as shown in the figures below. The solar panel is laid flat on top of the cart. This position is for transportation and storage as shown in the figures below.

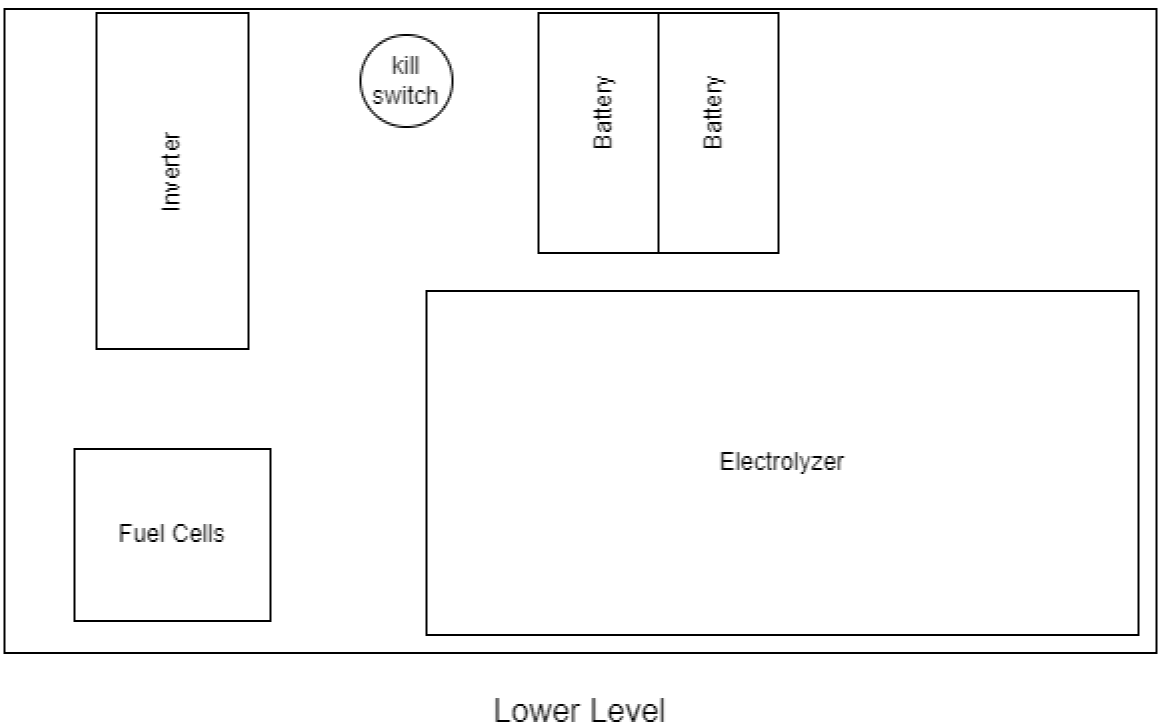


Figure. Layout for the lower level of the cart

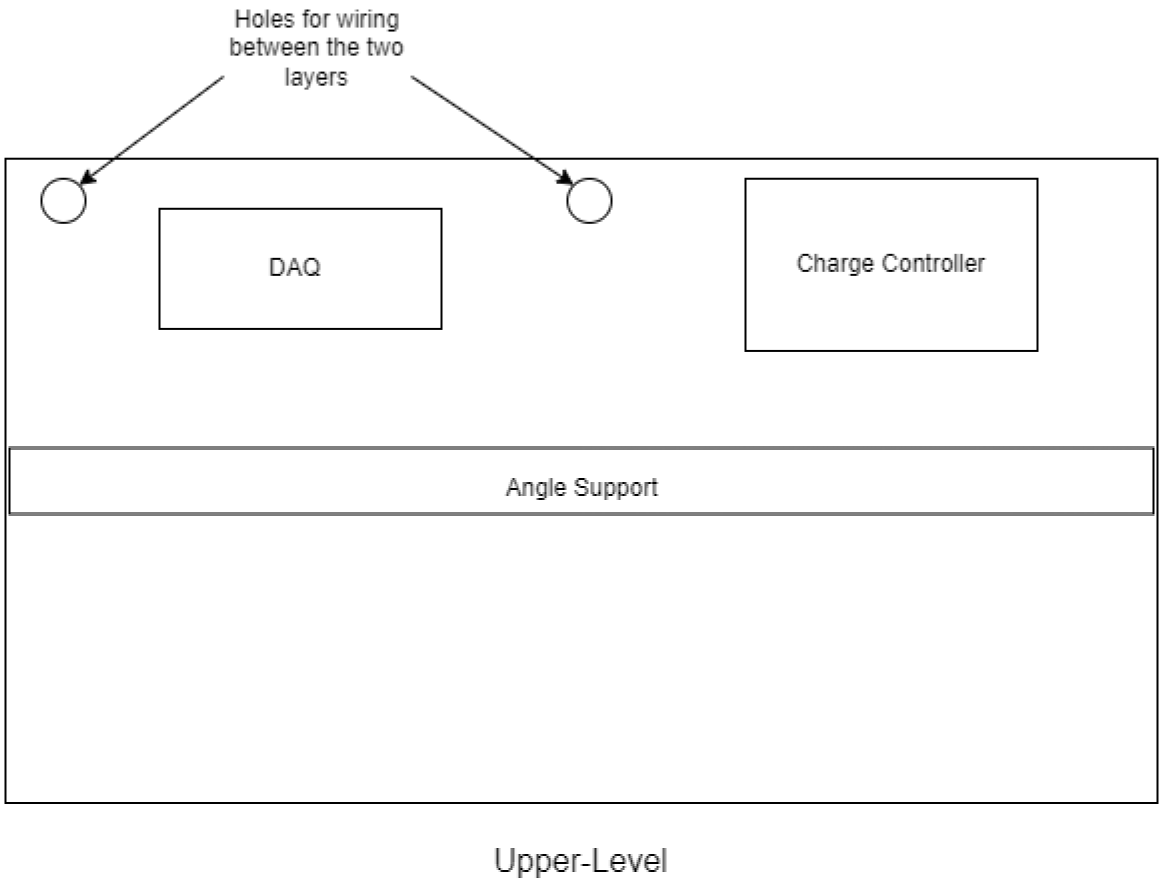


Figure. Layout for the upper level of the cart

Cart Assembly:

Hinge Attachment:

- Two metal hinges are used to attach the solar panel to the cart.
- The hinges are mounted on a wooden block, which is then fixed to the cart.
- Screws are used to attach the metal hinges to the wooden block. These screws are driven through the hinge plates into the wood, ensuring a robust connection.
- Additional screws are used to fasten the hinges to the solar panel's frame. This ensures that the panel is securely held in place and can pivot around the hinge axis.
- The hinge mechanism allows the solar panel to be tilted and adjusted to capture the maximum amount of sunlight.

Figure 5: Solar panel hinge assembly with self-tapping screws

Figure 6: Inside hinge view under solar panel screws

Solar Panel Installation:

- The solar panel is mounted at an angle on a triangular frame structure.
- The panel is supported by a wooden frame, which is attached to the cart.
- A wooden plank provides support to the solar panel and maintains the angle of the solar panel.
- This ensures the panel remains in an optimal position to capture sunlight effectively

Figure 7: Wooden panel support structure

- Solar panel is attached to a horizontal wooden plank
- Screws are used to attach the wooden planks to the solar panel. These screws are driven into the wood to the solar panel, ensuring a robust connection.

- The horizontal wooden plank is attached to a vertical wooden plank with a metal hinge
- The metal hinge is securely fastened with multiple screws.

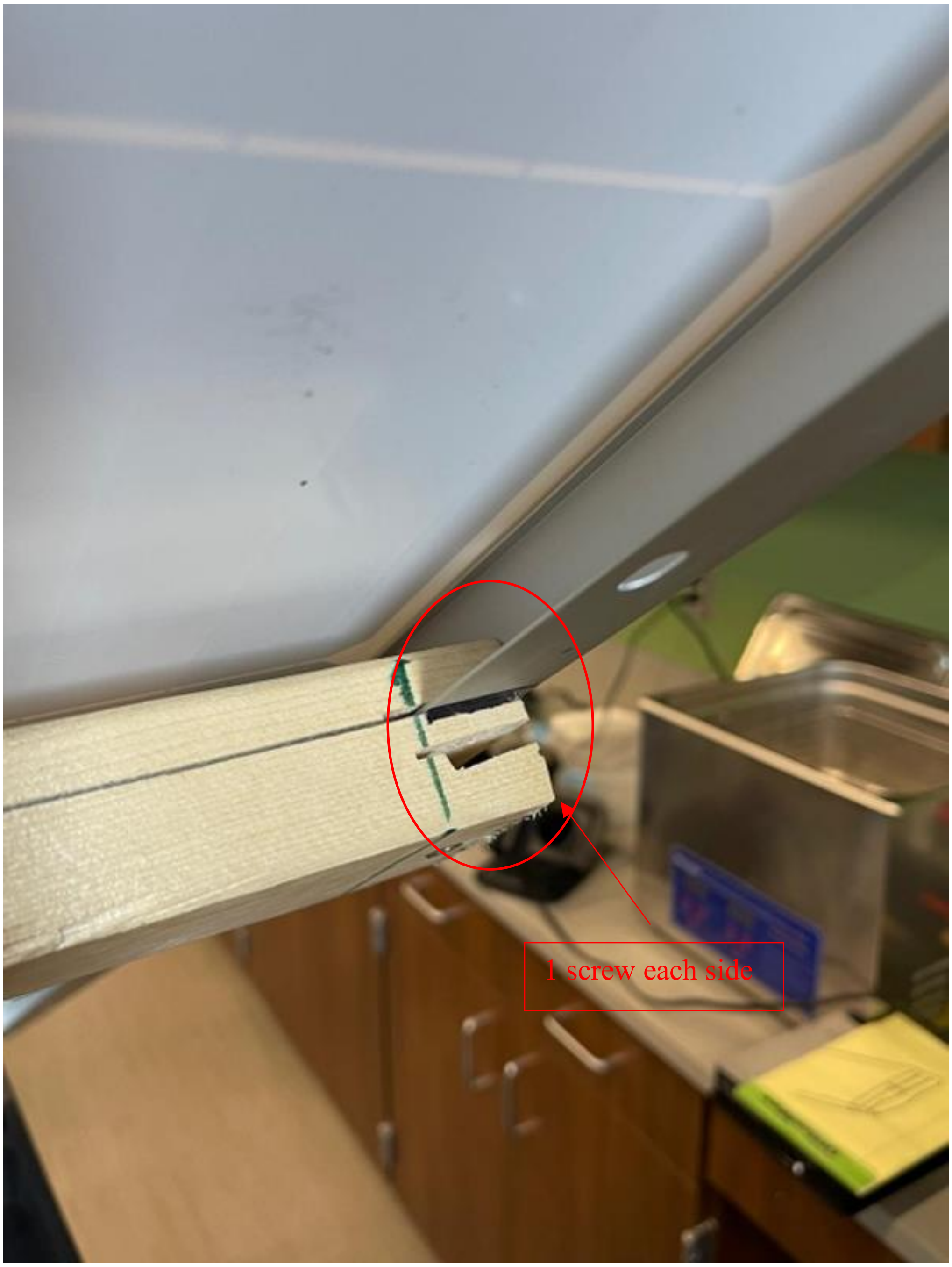


Figure 8: Horizontal wood plank is slid in the groove and mounted to the solar panel using 1 screw on each side

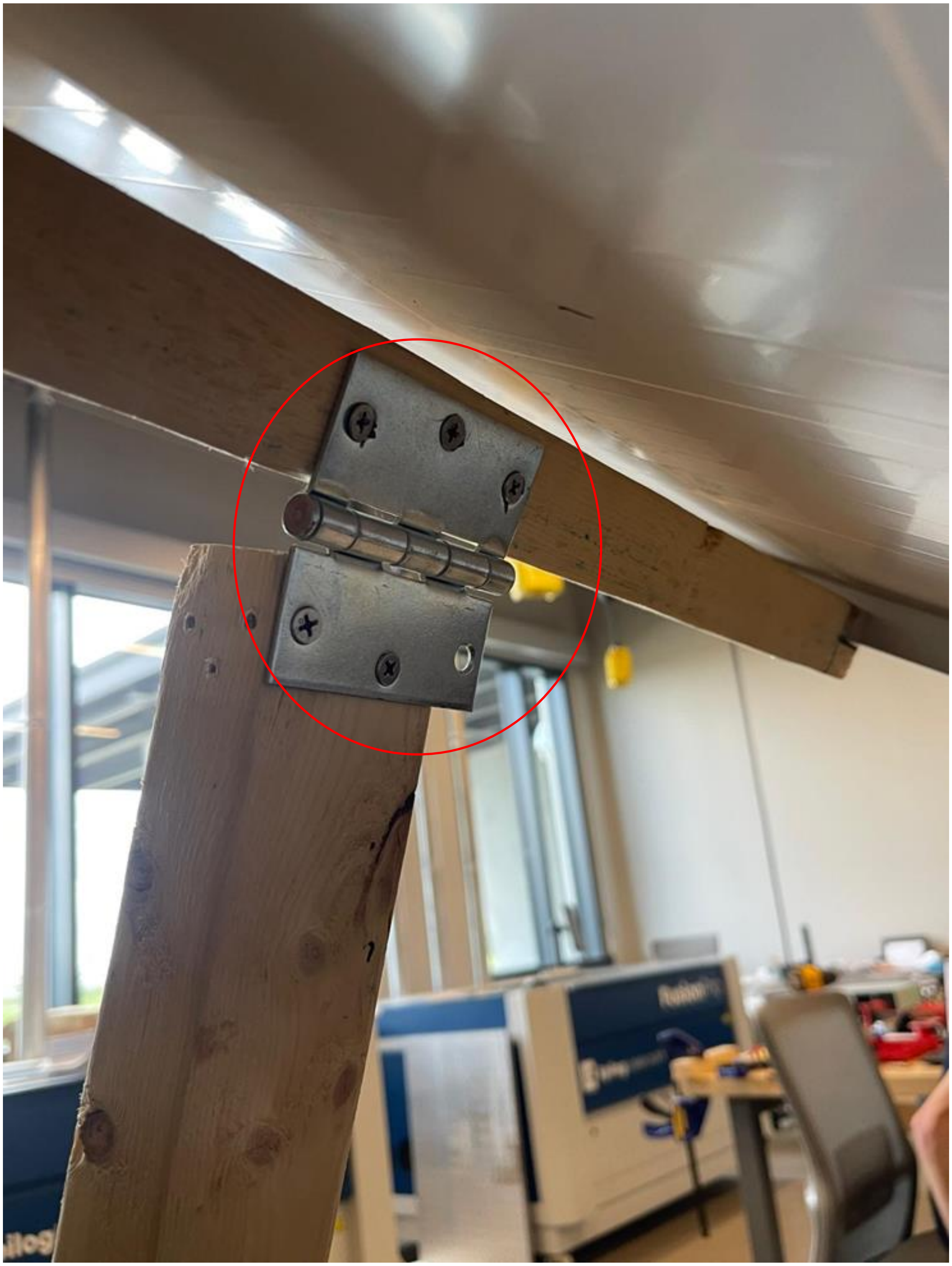


Figure 9: The vertical wood plank is mounted with hinge and screws.

Figure 10: The wood Structure is attached to the panel and rest on the inside cart edge fully extended to 45 degrees.

Figure 11: Using 2 screws on each mount, mount the angle brackets

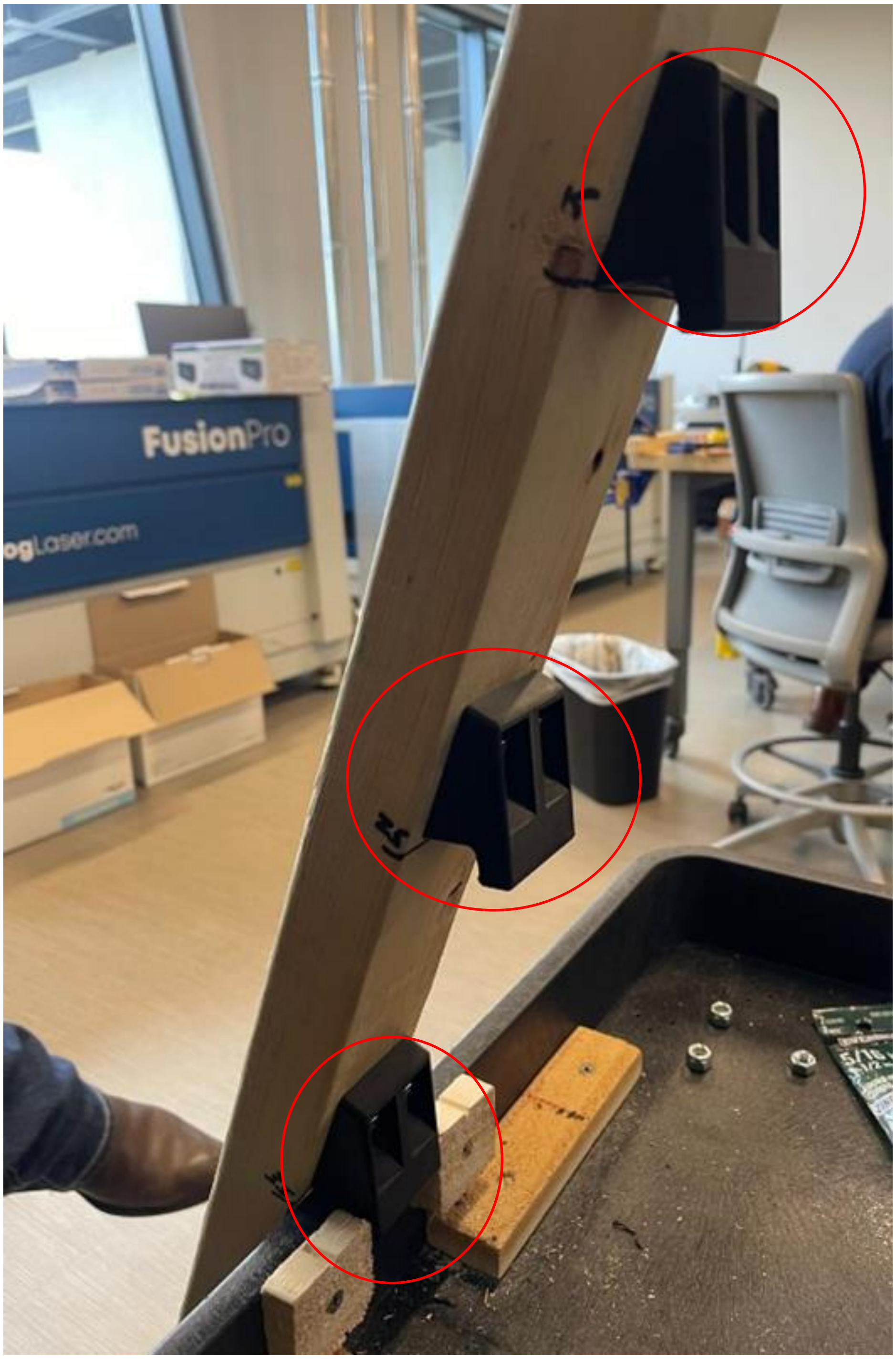


Figure 12: 3D Printed angle mounts are placed on 10 degree increments

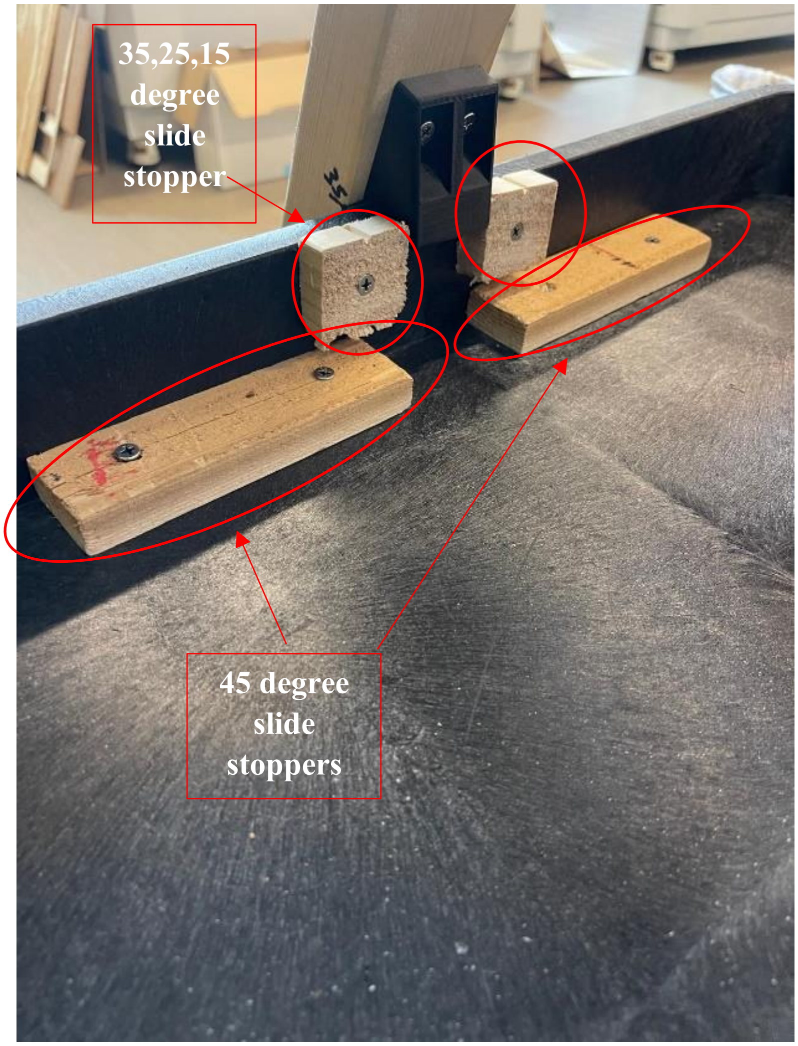


Figure 13: Wood Pieces are placed to stop the vertical angle from moving side to side

Charge Controller installation:

Battery & inverter installation:

Hydrogen fuel cells system assembly:

Operating Instructions: (Please read carefully before starting):
Warnings:
1. The tube between the hydrogen pressure regulator and the fuel cell gas input is required to be less than 30cm. The inner diameter of the hydrogen supply tube is required to be more than 2mm. The input pressure to the stack is required to be 0.45-0.55Bar.
2. Disconnect the hydrogen tube from the hydrogen input immediately after the fuel cell stack is shut down. Since hydrogen gas can leak into the fuel cell and destroy the stack.

3. Make sure the dry Hydrogen gas to be used must be≥99.995% purity.
4. Do not vibrate the stack when it is in operation.
5. Keep the stack in ventilation when it is in operation.
6. Keep the hydrogen outlet joint horizontal and on the bottom.
7. The tube between stack output and purging valve is required to be less than 20cm. The tube connected to the purging valve output is required to be less than 30cm. The inner diameter of the tube is required to be more than 2mm.

Step 1: Use the tube to connect the hydrogen supply (0.45-0.55 Bar) to the fuel cell hydrogen input port. Connect the controller to the fuel cell and also connect the controller to the purging valve.

Note: The white plastic point of the purging valve is for output and the metal point for input.
Warning: The tube between the hydrogen pressure regulator and the fuel cell gas input is required to be less than 30cm.

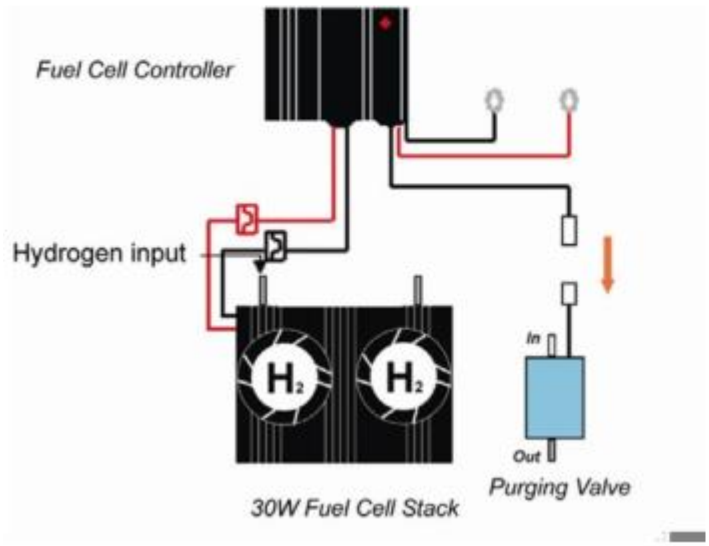


Figure. Dflsfsfsdp;asd,opkpoasg

Step 2: After the hydrogen (0.45-0.55 Bar) is supplied, the fan/blower should start to operate. The hydrogen shall flow through the stack. The red light on the controller should be on for a short time and then it will go off. Connect a tube to the purging valve input.

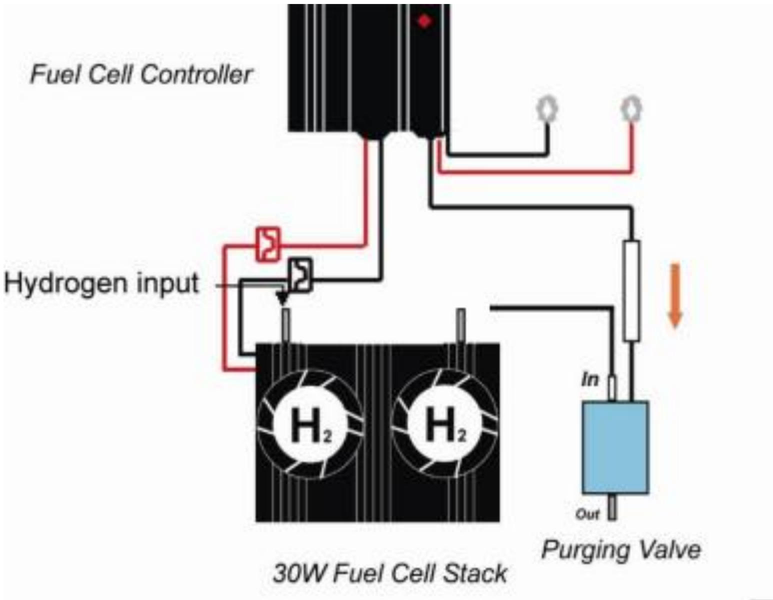


Figure. Dflp;asd,opkposasasg

Step 3: To obtain higher efficiency and better water management, please connect the other end of the tube from the purging valve to the hydrogen outlet port.

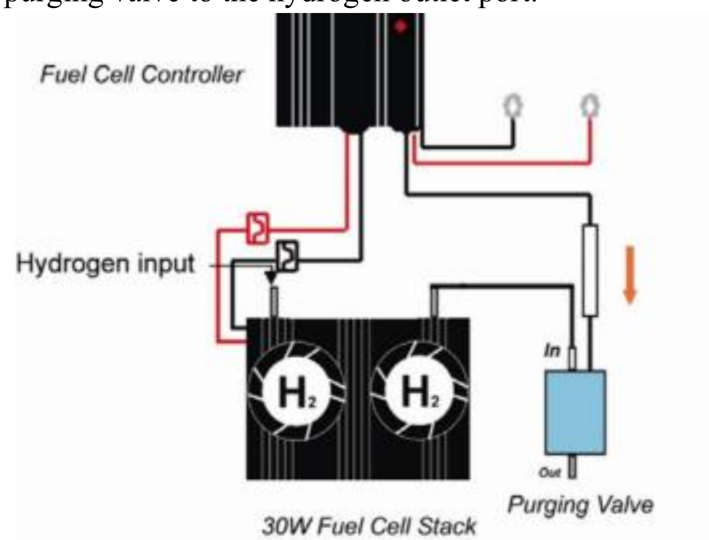


Figure. Dflp;asd,opkpoasasasdg

Step 4: The fuel cell system is now ready for use. Please connect the load to the fuel cell system.

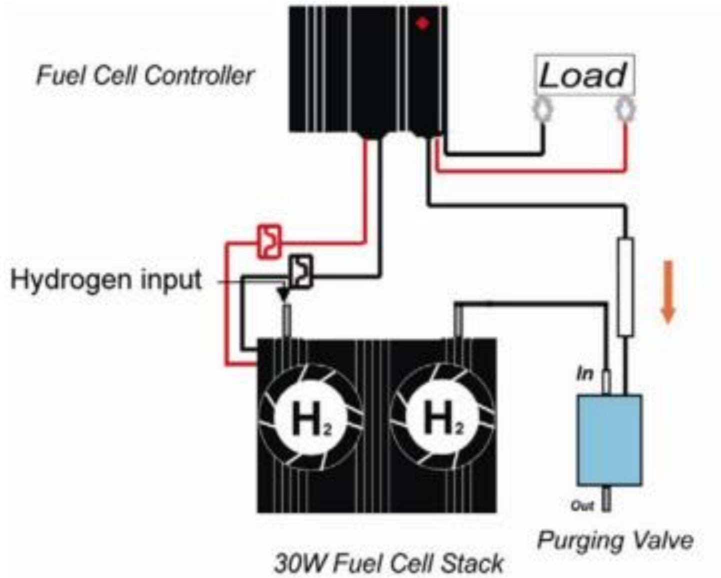


Figure. Dflp;asd,opkpoasgdasd

WARNING: Disconnect the hydrogen tube from the hydrogen input port immediately after the fuel cell stack is shut down. Since hydrogen gas can leak into the fuel cell and destroy the stack.

Simplified drawings of HFCT measurement stand

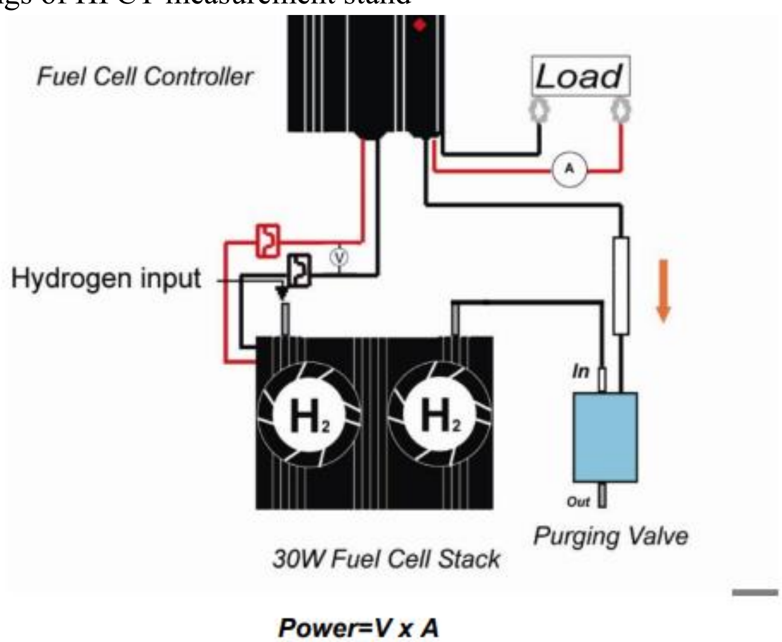


Figure. Dflp;asd,adsfcasfssaasdfs

Electrical components Assembly:

i. Connect the solar PV to the Charge controller:

Figure

Figure

ii. Charge controller to the batteries:

iii. Batteries to the invertor:

Mention the

iv. Invertor to the load:

v. Electrolyzer and FC:

DAQ system setup:

Sensor's setup and deployment:

Cloud Setup and Operation:

Steps for cloud data acquisition of Environmental Sensors via Mbed Studio, DISCO-IoT Board & Microsoft Azure

First Phase: Testing Connection to Azure with Mbed and DISCO-IOT101 Board

1. Azure Account setup: (personal + student($100 credit))
    a. Sign Up for Microsoft Azure, gives free credits for the first time sign-up on your name.
2. MBED studio:
    a. Go to the mbed studio, download from their website if you don't have it yet, click on a new project, select mbed-os-example-for-azure to create the test connection project.

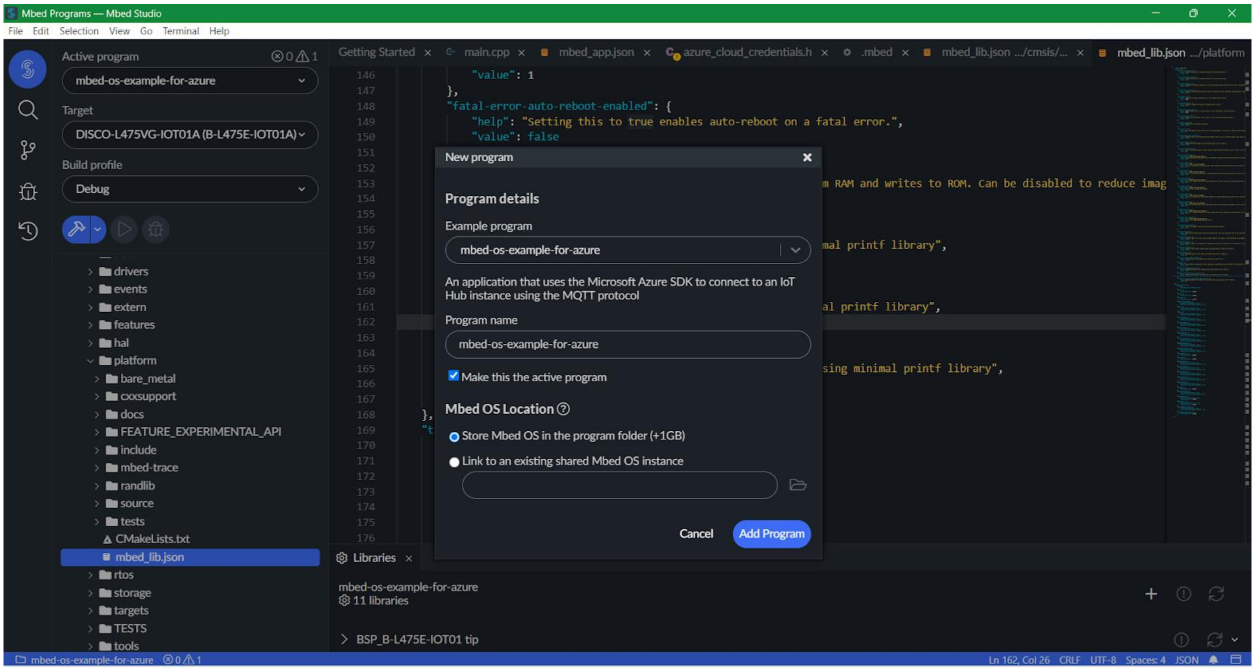


Figure n. Creating a new project in MBED using mbed-os-example-for-azure.

b. After the the full creation of the files (automatically creates the file structure, takes up to 2mins), from your file structure pane on the left, go to /c:/Users/your_user_name/Mbed Programs/mbed-os-example-for-azure/mbed-os/platform/mbed_lib.json
c. Inside the file mbed_lib.json find the json key with the name minimal-printf-enable-floating-point and overwrite the value true if it's false.

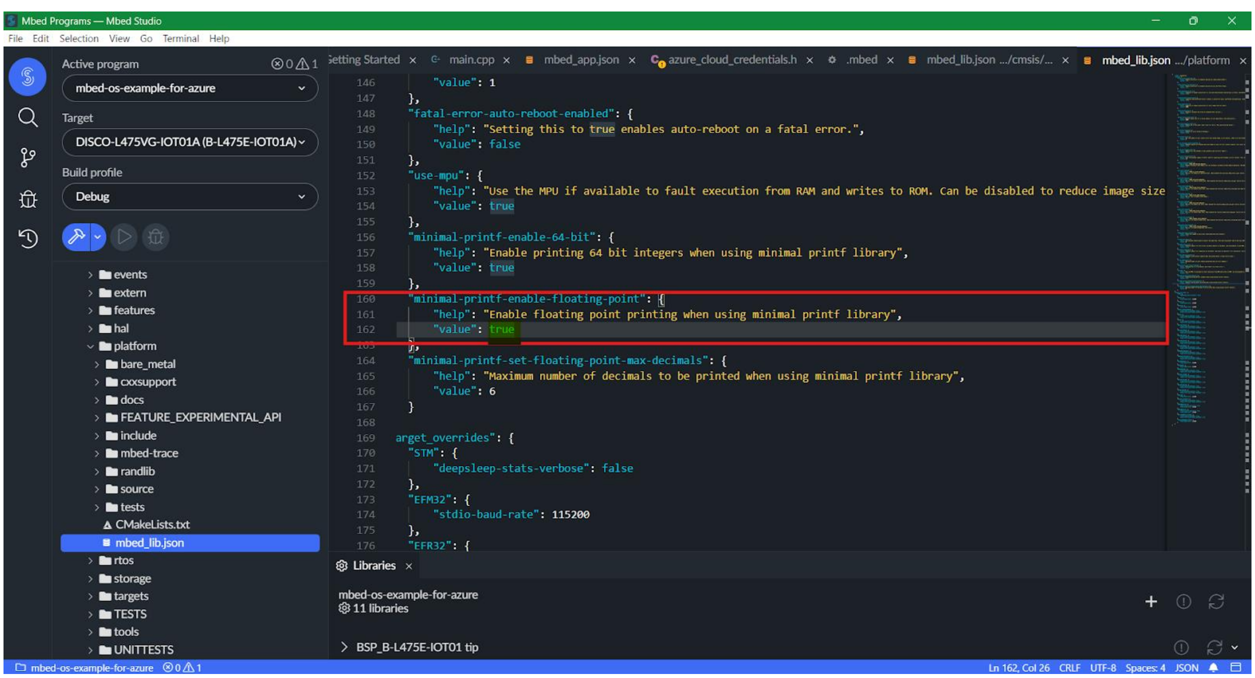


Figure. Enabling floating point printing in the mbed_app.json file

c. Click on File tab at the top left and click on “Add Library to Active Program”. Add https://os.mbed.com/teams/ST/code/BSP_B-L475E-IOT01/ as a library to the active project:

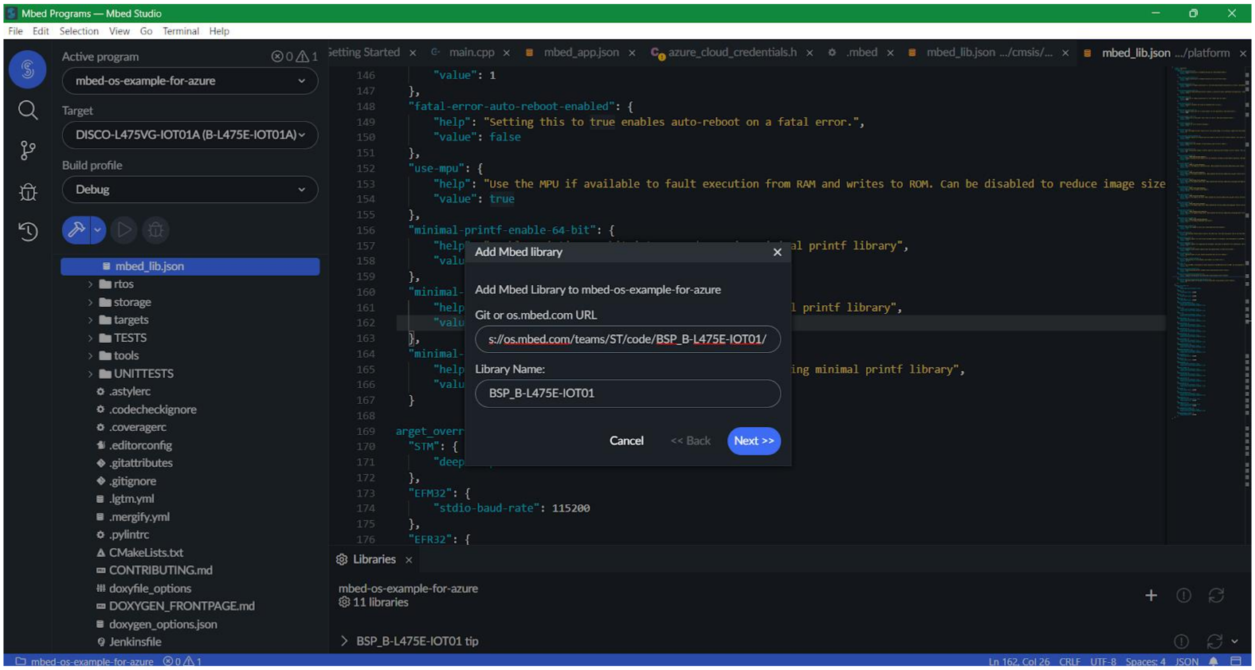


Figure n: Adding the BSP_B-L475E-IOT01A library to the Mbed project using a Git URL

d. Wi-Fi setup: Customize mbed_app.json for your Wi-Fi connection (screenshot) (Wi-Fi should be private; requiring a password; do not use a public connection) We know that the DISCO-IOT Board supports WAN and WAN2 so, I have used the hotspot of my mobile phone in this phase, in our observation, the Board couldn’t connect a public Wi-Fi that didn’t require a password.

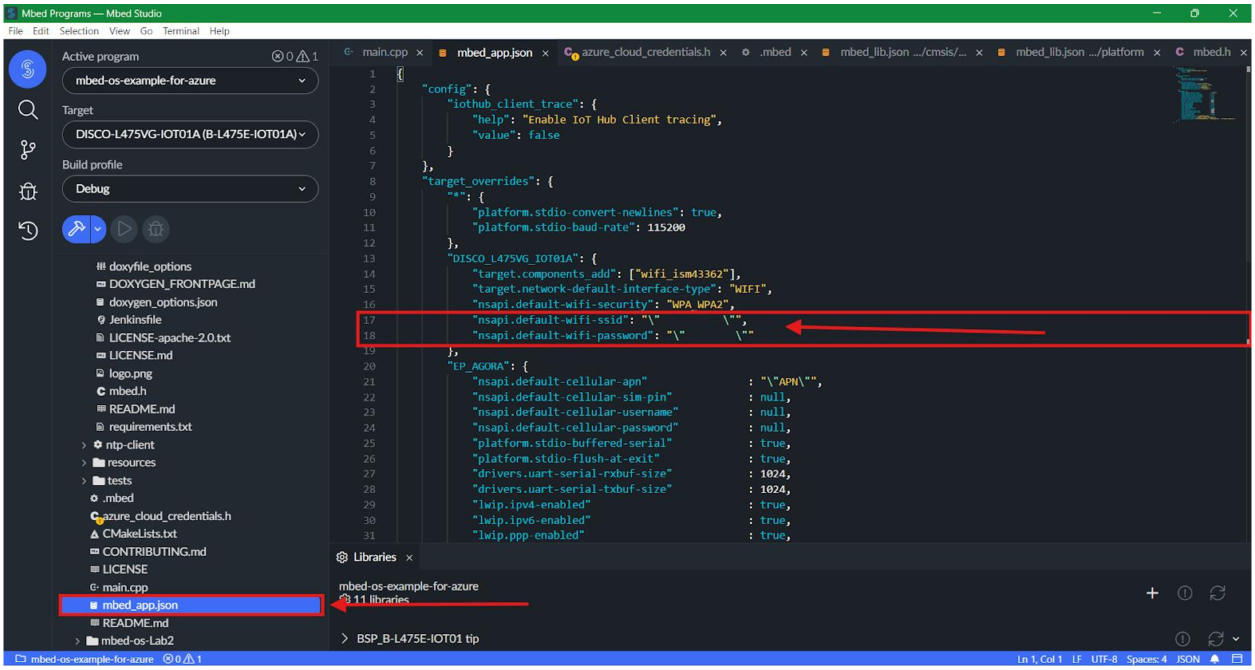


Figure X. Configuring Wi-Fi SSID and password in mbed_app.json for the DISCO-L475VG-IOT01A board.

3. In this step we will create an IoT Hub in the Azure Portal and in the IoT Hub, we will define a "Device" address for our DISCO-IoT Board and copy the primary connection string inside our mbed_app.json.
   a. Go to Azure Portal and Click on Create a resource. You will create a new IoT Hub Service.

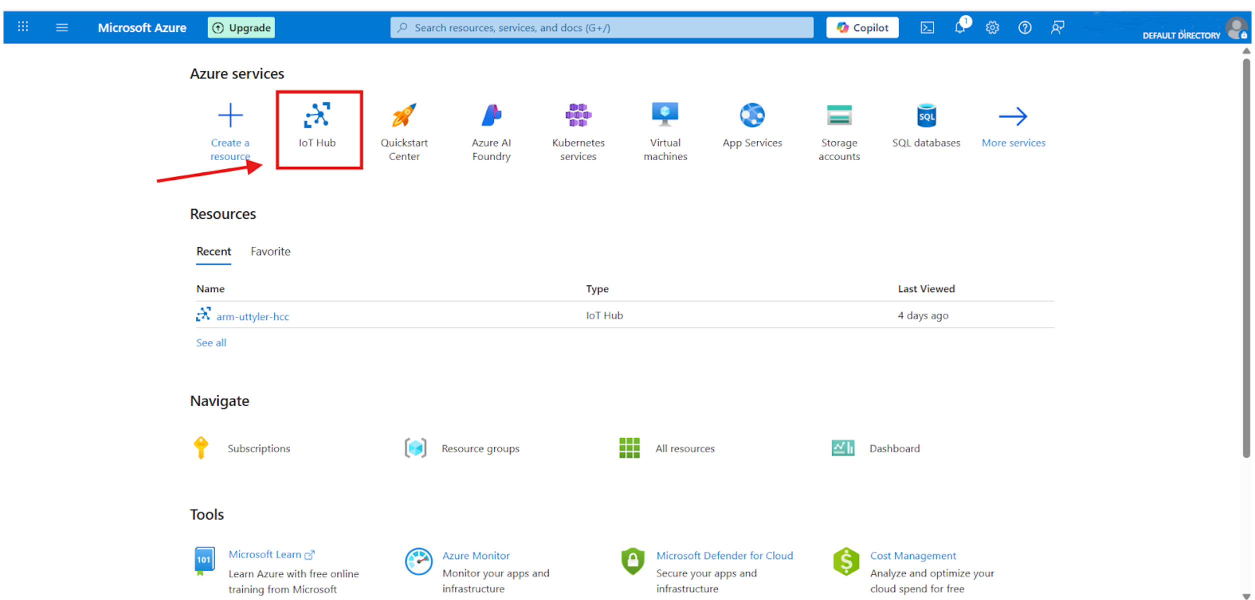


Figure X. Accessing the IoT Hub service in the Microsoft Azure portal.

   b. Click on your new service that you have created.

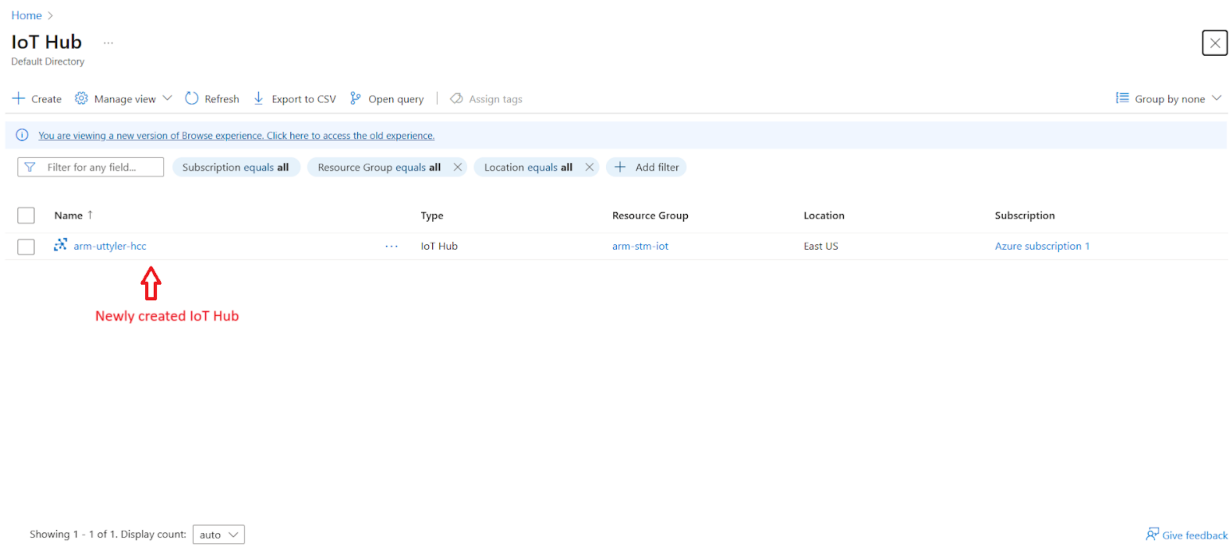


Figure X. Confirmation of newly created IoT Hub instance in the Azure portal.

c. Search "Devices" on the searchbar and create a new device.

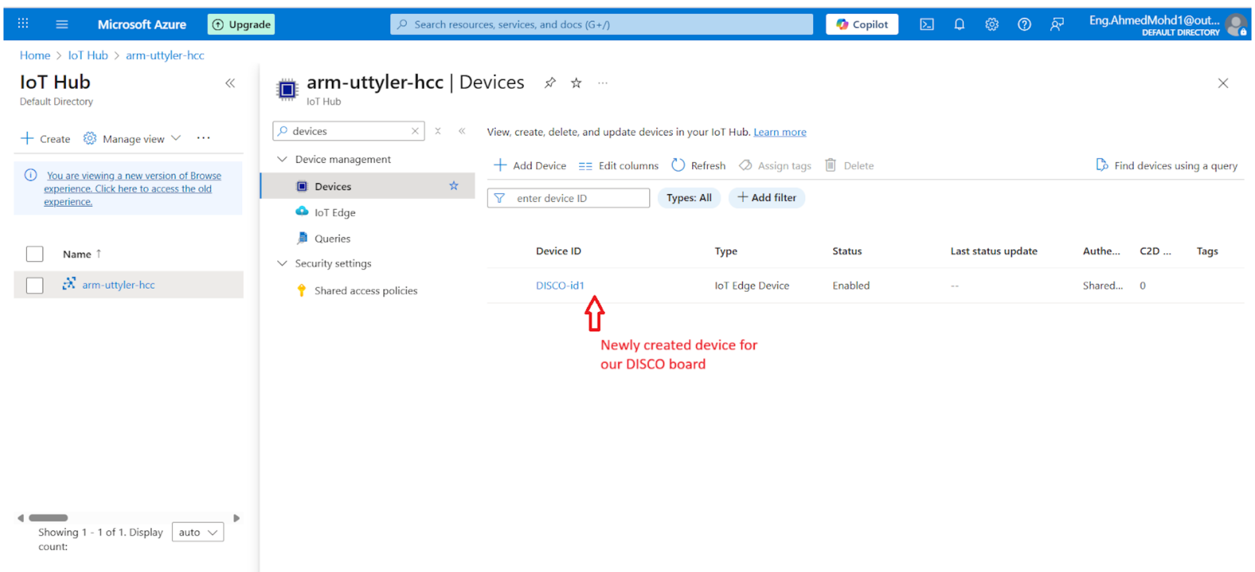


Figure X. Device registration in IoT Hub for the DISCO-L475VG IoT01A board

d. Copy the Primary connection string and paste it to your azure_cloud_credentials.h file. Remember, this key is a private key and shouldn't be exposed on the web.

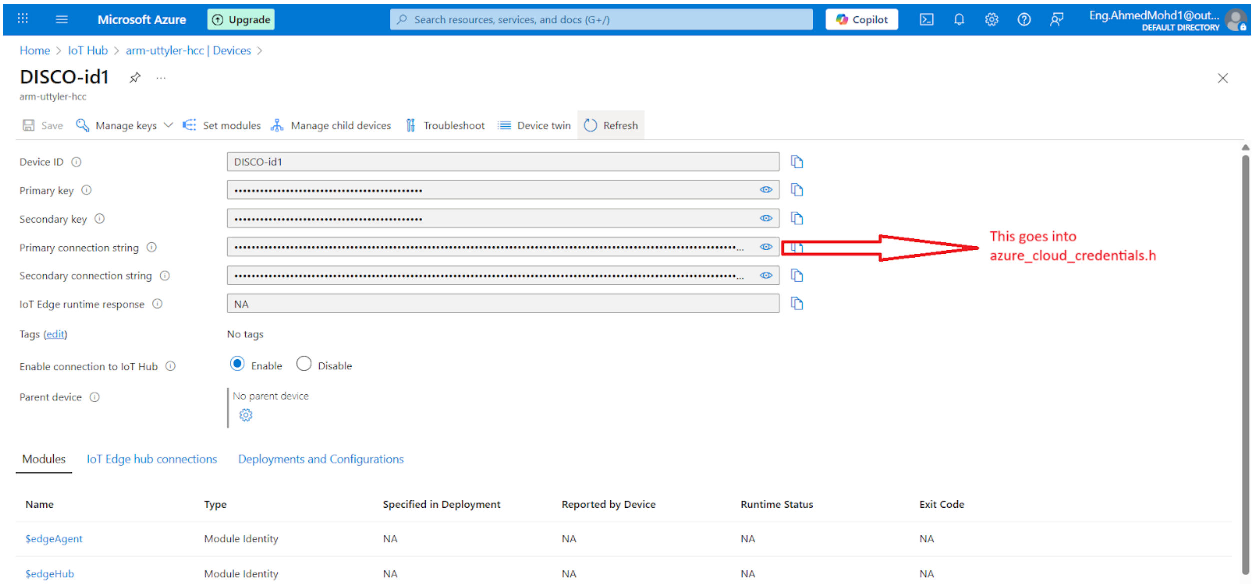


Figure X. Retrieving the primary connection string for the registered IoT device.

4. Now that your device is defined in an address in the cloud, you can share the values that are printed in the Mbed Studio terminal to the Azure as a payload. Remember, in this phase we are just testing the telemetry from the board. Connect the DISCO-IOT101 device to the laptop and build&run the program. You should be able to see on the device terminal in mbed studio that it has successfully connected to the Cloud and sent the messages.

5. To make sure the messages went through, we have connected the azure cloud from windows powershell also and displayed the messages in json format from the Azure Cloud's end this time, not from the device's end in Mbed Studio. Follow the steps below to do so:(This is optional, if you are seeing messages in the Mbed Studio terminal this is unnecessary)
    a. Assuming you are using a Windows OS, open your powershell and copy paste the command: winget install –exact –id Microsoft.AzureCLI
    b. Check the az –version
    c. "C:\Program Files\Microsoft SDKs\Azure\CLI2\python.exe" -m pip install –upgrade pip setuptools (Configure the directory path according to your pc setup)
    d. Run in the terminal: $Env:PYTHONPATH = 'C:\Program Files\Microsoft SDKs\Azure\CLI2"
    e. Run in the terminal: az extension add –name azure-iot –upgrade
    f. Run in the terminal: az login (Log in to your Azure account created in step 1)
    g. az iot hub monitor-events –hub-name arm-uttyler-hcc –device-id DISCO-id1 (Again configure the device name as the name you created in step 3)
    h. Connect your board and run the program on Mbed Studio
    i. Check the Microsoft PowerShell if you are receiving messages from the board on the cloud's end.

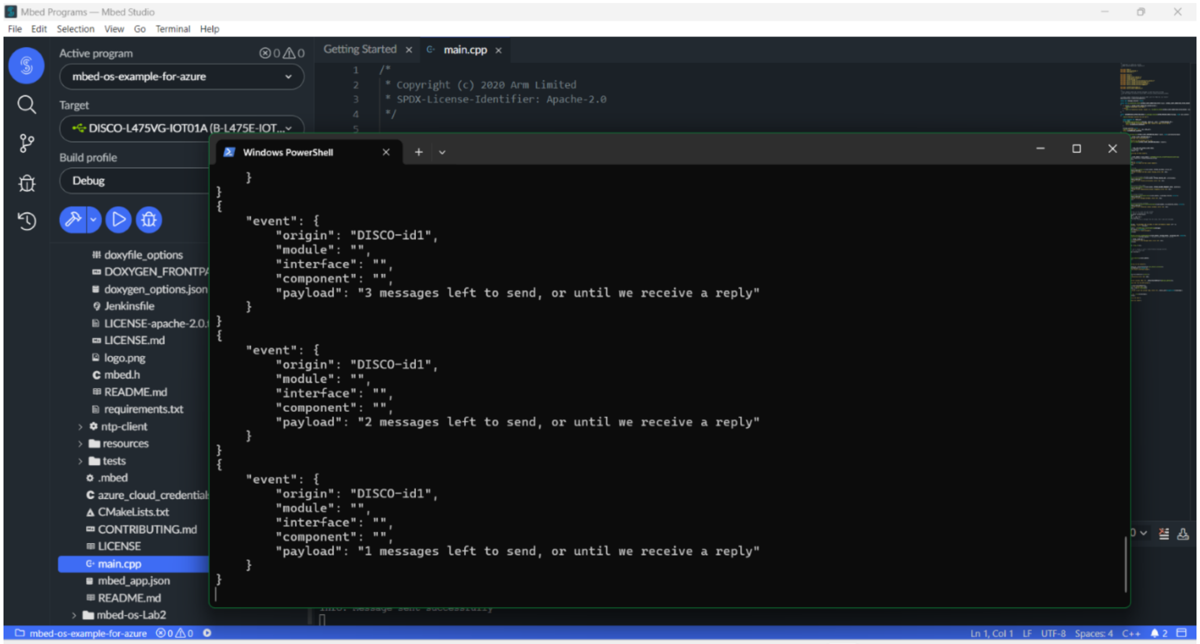


Figure X. Monitoring device-to-cloud messages using Azure CLI in Windows PowerShell.

Second Phase: Sending the Environmental Sensor Telemetry to Azure(No Database Creation yet)

1. In this phase, instead of sending a custom message, we will send actual real-time sensor data from our board, but we won't save it to our database yet, we are testing if our sensor readings are correct. A main.cpp file will be shared in the file structure which has the code to read the sensor values, all you need to do is to copy the content and paste it to the original main.cpp from the first phase. You will build and run the program again and you should be exploring to see the sensor readings from your Mbed Studio.

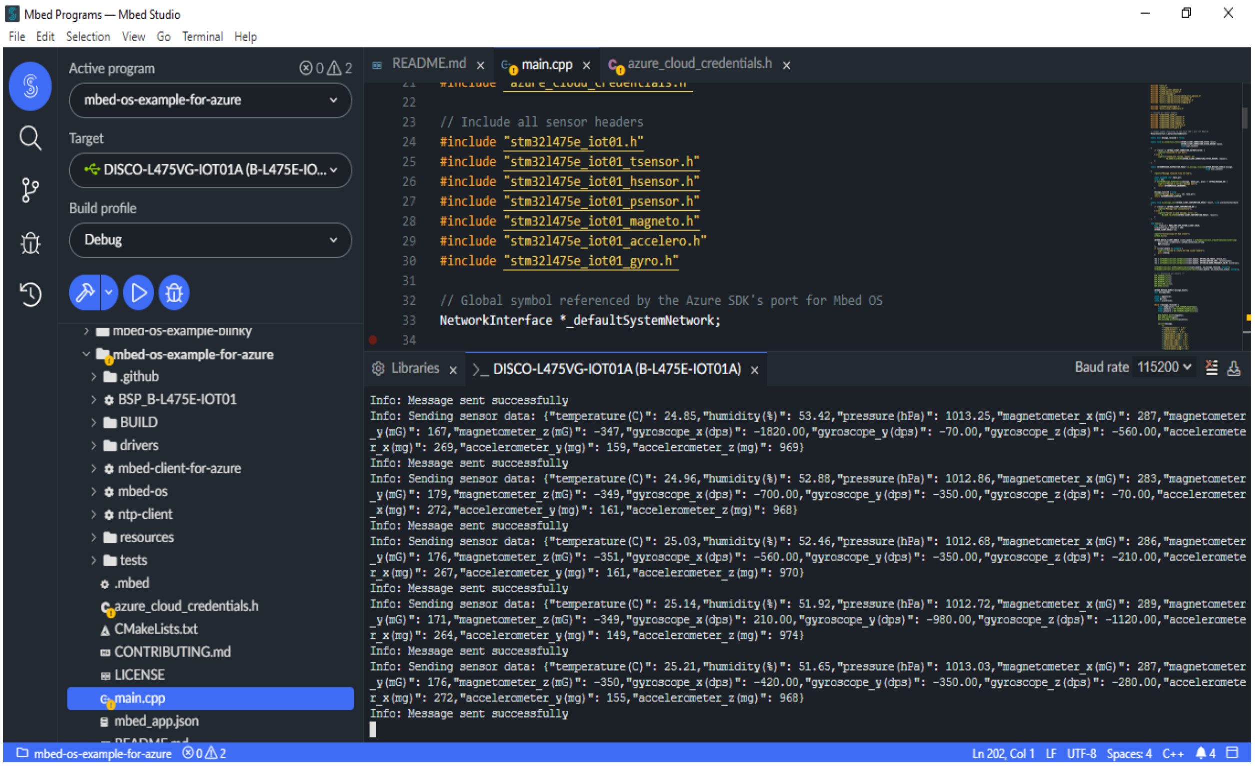


Figure X. Real-time sensor data transmission from DISCO-L475VG-IOT01A displayed in Mbed Studio

Third Phase: Storing NoSQL Data in Azure CosmosDB from Live Sensor Telemetry of DISCO-IOT Board

This is the phase that we are completing our Cloud Data Pipeline. From the flowchart below, you will explore how the whole setup should be. In summary, we will create a CosmosDB Service, configure our Database and Container in it for the data to be stored in that address. After, we will create a Custom Endpoint and Route for the Live Telemetry Messages to be routed to our CosmosDB Container address.

1. Go to Azure Portal and click on Create a resource. You will then find the CosmosDB for NoSQL, click on it to create the service. Don't forget to set it to serverless. I have configured the name of the CosmosDB as myiotcosmosdb for the project, so you will see that name for the service in the images below.

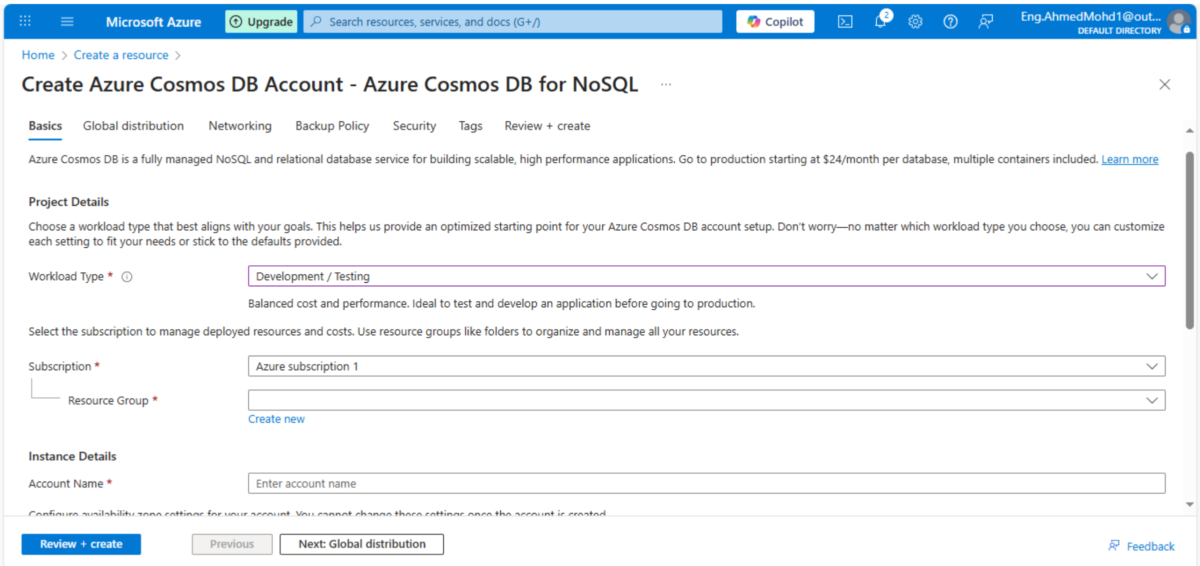


Figure X. Initial setup for creating an Azure Cosmos DB account for NoSQL in the Azure Portal.

2. After the creation, go to your CosmosDB Service > Click on the name of the newly created DB(myiotcosmosdb for me) > Search on the searchbar and click on Data Explorer > Create new Container > Assign id to your newly created database(mine is IoTData) > Assign id to your container(mine is Telemetry) > Put /DeviceId as partition key and click create.

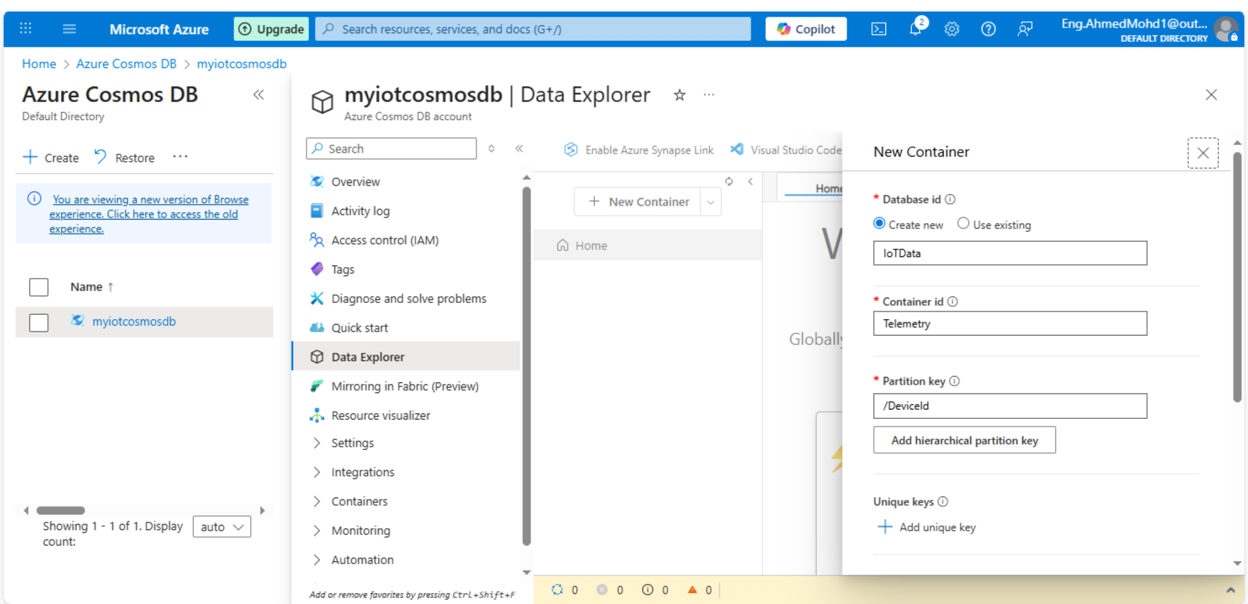


Figure X. Creating a new database and container in Azure Cosmos DB using Data Explorer

3. Now that we have our Database, only thing left to do is to configure an "Endpoint" and a "Route" for our Cloud Service to create a connection for the data transfer from our defined "Device" in the IoT Hub to our Container called "Telemetry" in the CosmosDB.

a. Firstly, go to IoT Hub from Azure Portal, select your service, search Message Routing in the sidebar and click on add. You will configure an "Endpoint" first. Don't forget to disable "Generate a synthetic partition key for messages"

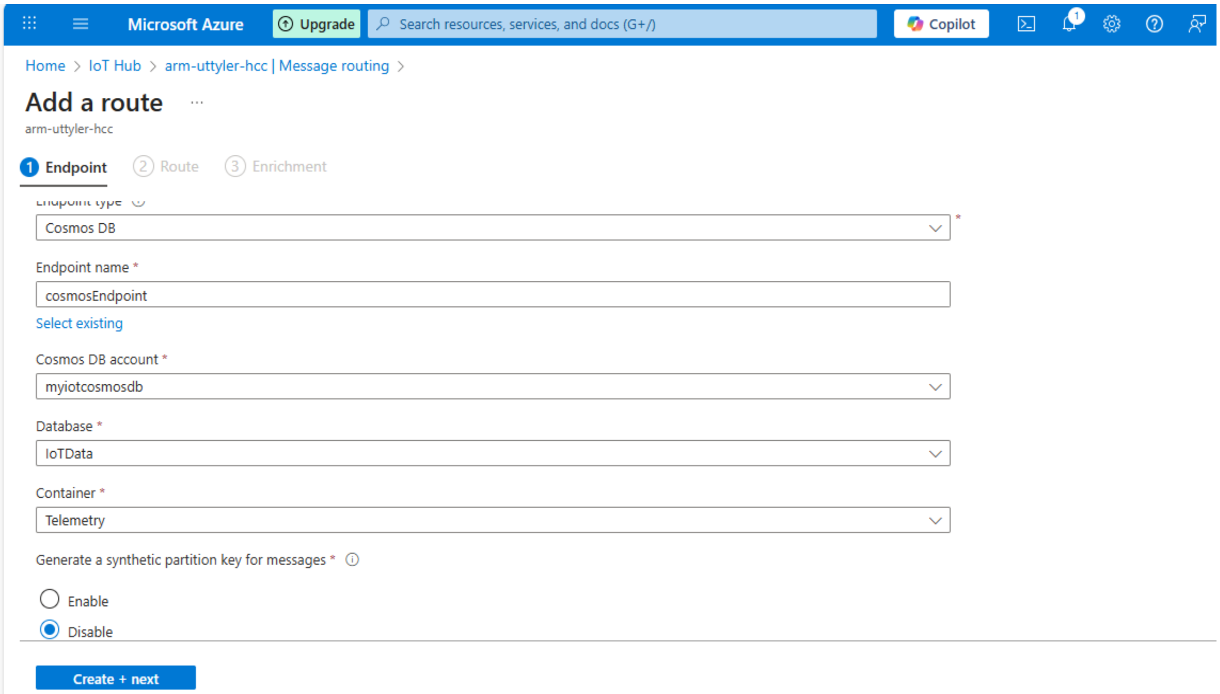


Figure X. Configuring a Cosmos DB endpoint for message routing in Azure IoT Hub.

b. After clicking on Create + next now you are creating a route. I named it telemetryToCosmos. Click on Create + skip enrichments.

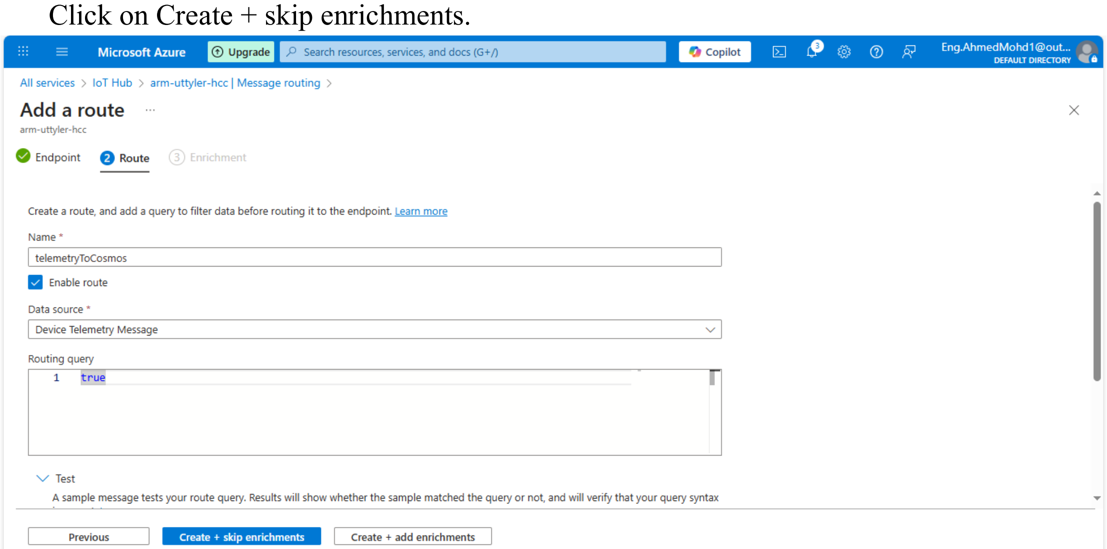


Figure X. Creating a route to forward telemetry messages to Cosmos DB.

4. We are almost there! Everything is set up and waiting for the system to run. Go to your CosmosDB > Click on Data Explorer > Click on IoTData > Telemetry > Items. Build and

run your board from Mbed Studio. After a couple of runs, hit refresh in the Data Explorer page. Congratulations! You should be able to see the new Telemetry data being saved every time you hit the refresh on the page because the is being transmitted every 5 seconds.

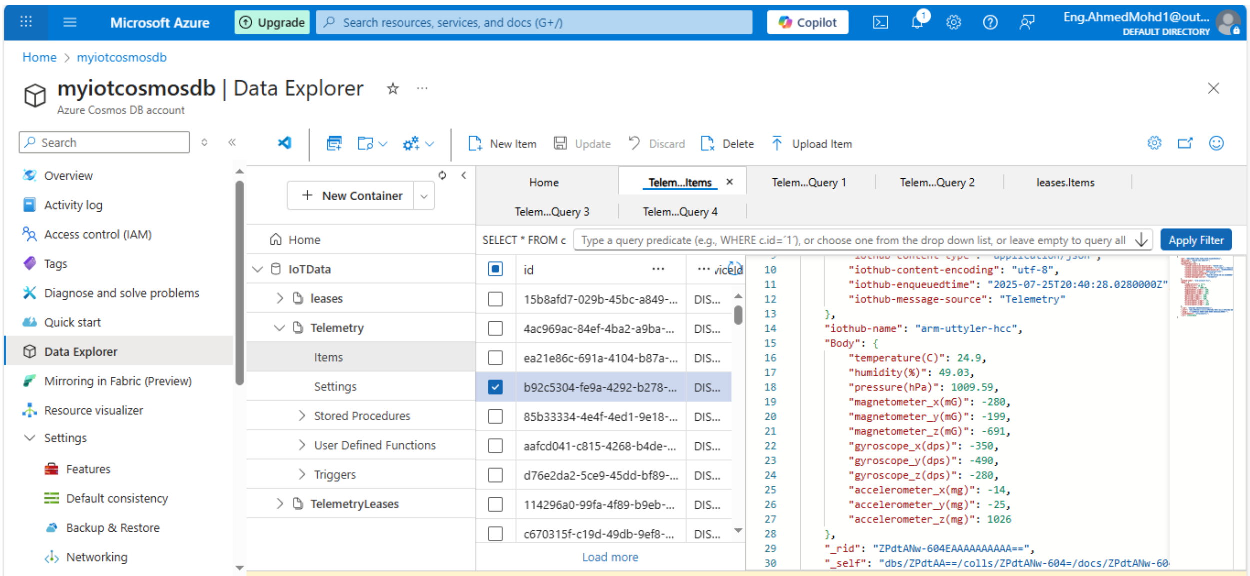


Figure X. Live telemetry data successfully stored in the Azure Cosmos DB container.

Operation and testing:

Charging the station:

i. Ensure Safety Before Moving the Cart:

Before transporting the solar charging station outdoors, make sure the kill switch is turned OFF to prevent accidental current flow during movement.

ii. Position the Solar Panel:

- Place the cart in an open area with direct sunlight.
- Adjust the solar panel to face due south (180°) for optimal exposure.
- Angle the panel to match local latitude (≈30° in Houston) to maximize energy capture, especially during 10 AM to 3 PM.
- Once the station is properly positioned and stabilized, turn ON the kill switch to enable current flow from the solar panel to the rest of the system

iii. Activate the Kill Switch:

Once the station is properly positioned and stabilized, turn ON the kill switch to enable current flow from the solar panel to the rest of the system.

iv. Solar Energy Generation Begins:
Under full sunlight, the Eco-Worthy 195W bifacial panel produces an open-circuit voltage (Voc) of approximately 20.7 V, consistent with its rated performance.

v. Power Routed Through Protection Circuit:

The generated DC power flows through a 15A DC circuit breaker (DZ47X-63), providing overcurrent protection to safeguard system components.

vi. Charge Controller Operation:
The Renogy MPPT charge controller receives the incoming power and regulates voltage and current to optimally charge the parallel-connected $LiFePO_4$ battery bank.

Figure . The solar charging station deployed outdoors with the bifacial solar panel oriented due south at an optimal tilt angle for maximum solar harvesting in Houston, TX.

Figure. A smartphone compass app is used to align the solar panel directly south (180°), ensuring proper orientation for maximum solar efficiency in the northern hemisphere.

Figure. A 15A DC circuit breaker (DIHOOL DZ47X-63) is installed between the solar panel and charge controller for overcurrent protection and system safety, it is switched on.

Solar Panel Output:

i. Use the arrow buttons on the Renogy Rover MPPT charge controller to navigate to the screen that displays the solar panel voltage. This screen confirms the voltage being received from the panel under sunlight.

ii. (Optional) to ensure accuracy, use a digital multimeter (set the multimeter to DC voltage measurement as seen in the figure) to measure the voltage at the PV input terminals of the charge controller. This step is helpful for verifying system performance or troubleshooting.

iii. Navigate through the charge controller's menu to locate the screen displaying the charging current (in Amps) being delivered to the battery bank. This confirms that the panel is actively charging the batteries.

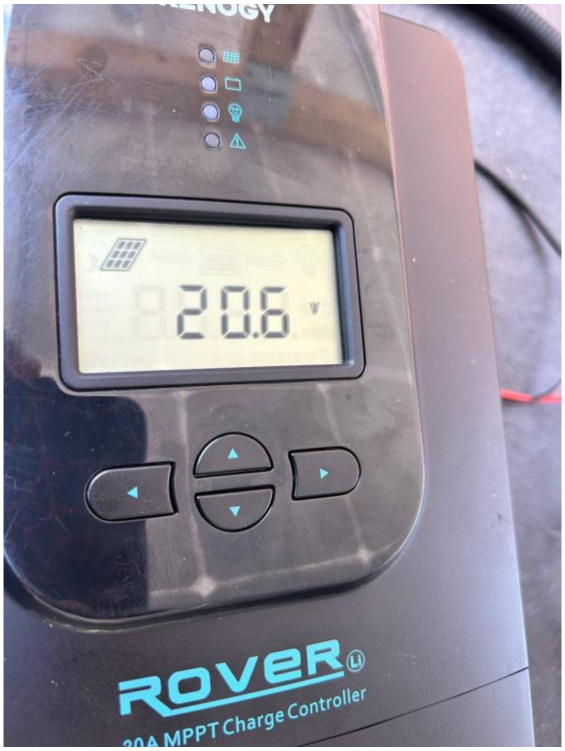


Figure. The Renogy Rover 20A MPPT charge controller display shows a real-time solar charging power of 20.6 watts, indicating the system is actively converting solar energy and delivering it to the battery bank.

Figure. The Renogy Rover MPPT charge controller display shows a charging current of 1.98 A, indicating the system is actively delivering current from the solar panel to the battery bank.

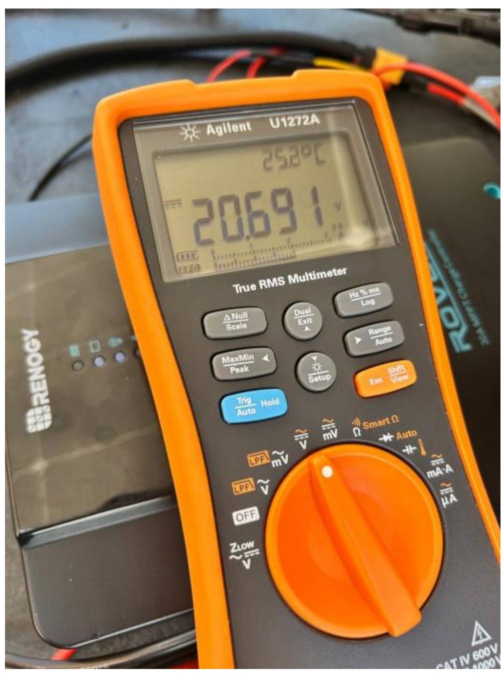


Figure Measurement of open-circuit voltage (Voc) of the solar panel under full sunlight using an Agilent U1272A multimeter. The reading shows 20.691 V, confirming proper panel output.

Charge Controller Output:

i. View battery voltage on the charge controller: Using the navigation buttons on the Renogy Rover MPPT charge controller, scroll to the screen that displays the battery voltage output. This reading shows the voltage currently being delivered to the battery bank.

ii. Confirm Battery Voltage Using a Multimeter: To validate the controller's reading, use a digital multimeter (set the multimeter to DC voltage measurement as seen in the figure) to measure the voltage directly at the battery terminals or at the charge controller's output terminals.

Figure. The Renogy Rover MPPT charge controller shows a battery charging voltage of 14.4 V, indicating that the system is in the BOOST phase and actively delivering power to the battery bank.

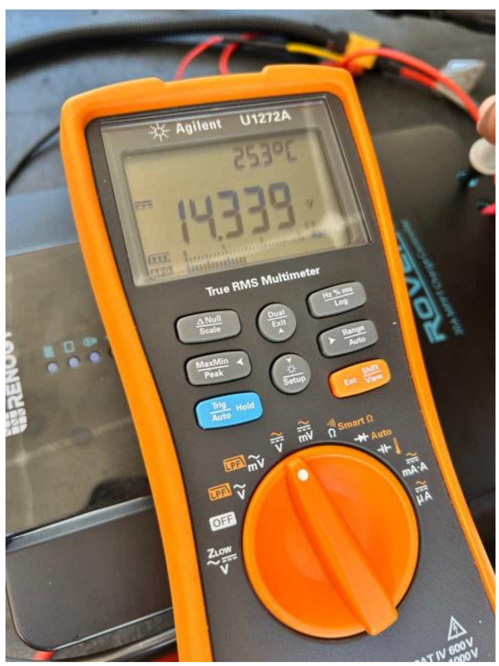


Figure. An Agilent U1272A multimeter confirms the charge controller's output by measuring a voltage of 14.339 V at the battery connection, validating that the controller is delivering appropriate charging voltage.

Battery Voltage:

i. Before connecting or powering any load, use a digital multimeter to measure the voltage across the terminals of the battery bank, set the multimeter to DC voltage measurement (as seen in the figure), place the red probe on the positive terminal and the black probe on the negative terminal.

ii. 13.334 V — the battery was being charged (charge controller supplying current).
12.961 V — the battery was in a discharging state (current flowing out to the inverter or another load). The voltage difference (~0.37 V) occurs because of internal resistance and voltage response to current direction.

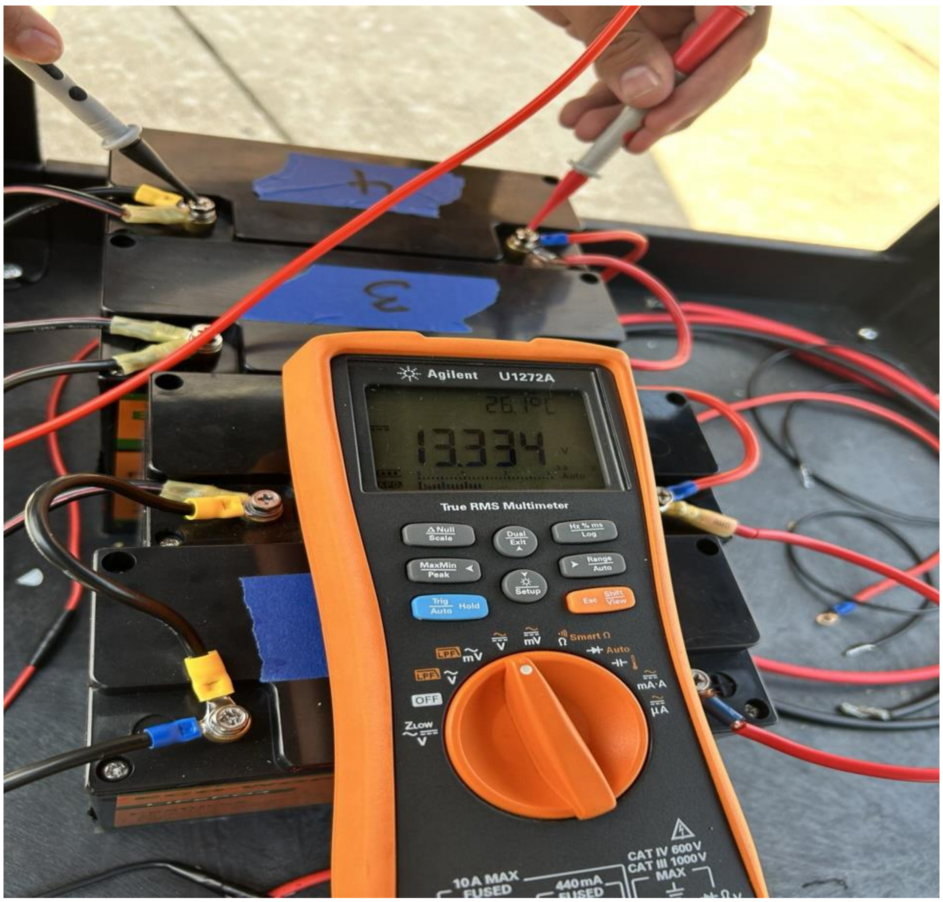


Figure. The multimeter shows 13.334 V, indicating the voltage under load conditions—slightly elevated due to active charging or recent current draw.

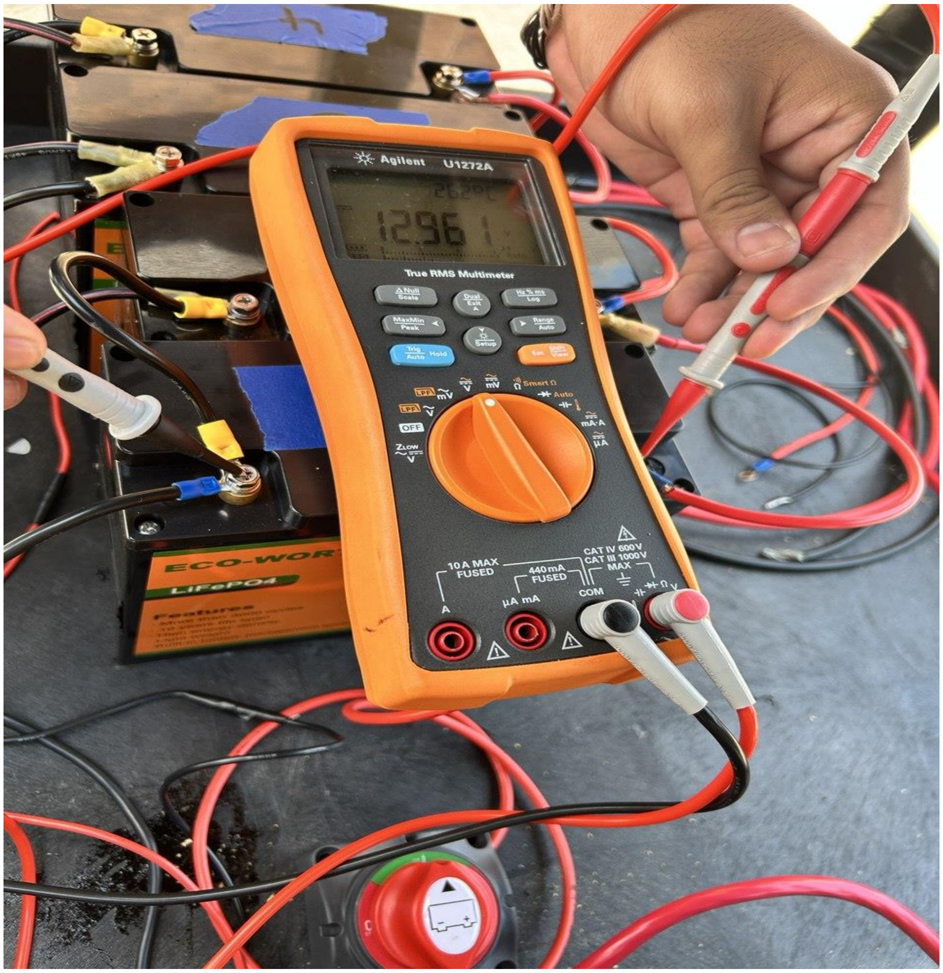


Figure. The multimeter reads 12.961 V, which represents the resting voltage of the battery bank under no active load.

Discharging (Using a load):

i. Turn ON the Inverter Power Switch: Locate the main switch (as seen in the figure) on the front panel of the inverter. Press the switch to the ON (|) position to enable power delivery from the battery bank.

ii. Connect and Begin Charging a Device: Plug a device (e.g., a smartphone) into the inverter's AC outlet or USB port. The device should begin charging, confirming that the inverter is successfully delivering AC power

iii. Check the inverter LCD display for real-time system metrics: DC Input Voltage from the battery bank AC Output Voltage to the load Load Power, Temperature, and Frequency.

Figure. The inverter is currently switched OFF, as indicated by the main rocker switch in the O position. Both the POWER and FAULT indicator LEDs are off. No AC power is being delivered to the output sockets or USB port in this state.

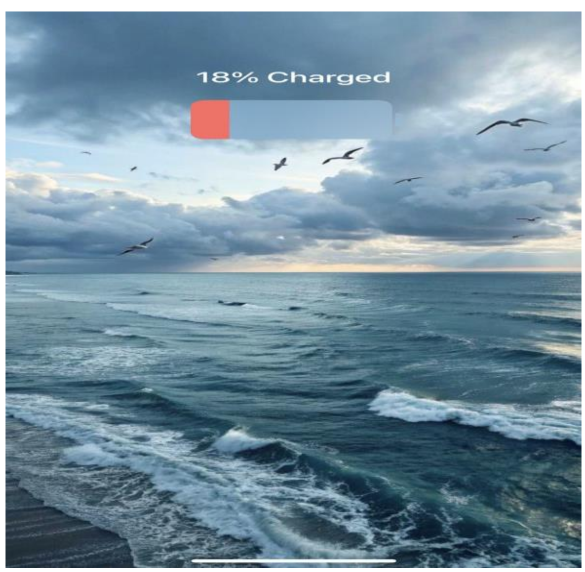


Figure. A connected device shows 18% charged, confirming successful AC output from the inverter to the load via USB or outlet connection.

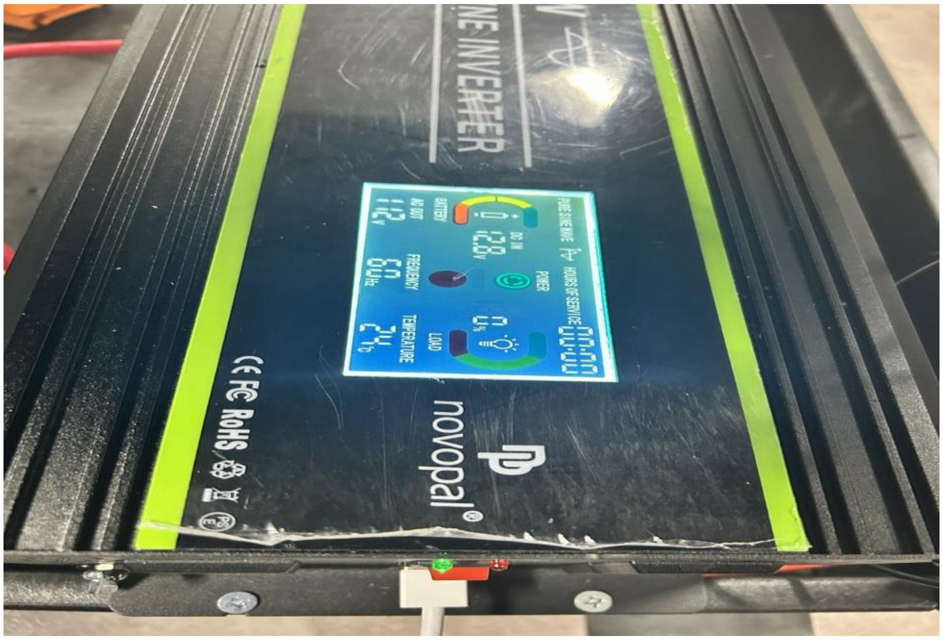


Figure. This indicates the inverter is powered on, inputting from the battery, and ready to supply clean AC power

Figure. The output panel of the inverter showing four AC outlets, USB port, and indicator lights. This is where devices are connected during battery discharging operations.

Full system deployment & testing:

Appendix A:

```
#include <Wire.h>
#include "Adafruit_MCP9601.h"

// === TCA9548A Multiplexer ===
```

```
#define TCA_ADDR 0x70

void selectTCA(uint8_t channel) {
  if (channel > 7) return;
  Wire.beginTransmission(TCA_ADDR);
  Wire.write(1 << channel);
  Wire.endTransmission();
  delay(100);  // Allow bus to stabilize
}

// === INA228 Configuration ===
#define NUM_INA 3
const uint8_t ina_addrs[NUM_INA] = {0x41, 0x44, 0x45};
#define REG_SHUNT_CAL   0x03
#define REG_VBUS        0x05
#define REG_CURRENT     0x07
#define SHUNT_RESISTANCE 0.0002
#define CURRENT_LSB      0.0001
#define VBUS_SCALE       62.0  // Calibrated multiplier

void writeINARegister16(uint8_t addr, uint8_t reg, uint16_t value) {
  Wire.beginTransmission(addr);
  Wire.write(reg);
  Wire.write(value >> 8);
  Wire.write(value & 0xFF);
  Wire.endTransmission();
}

int32_t readINARegister24(uint8_t addr, uint8_t reg) {
  Wire.beginTransmission(addr);
  Wire.write(reg);
  Wire.endTransmission(false);
  Wire.requestFrom(addr, (uint8_t)3);
  if (Wire.available() < 3) return 0;
  int32_t value = 0;
  value |= ((uint32_t)Wire.read()) << 16;
  value |= ((uint16_t)Wire.read()) << 8;
  value |= Wire.read();
  return value;
}

// === MCP9601 Configuration ===
#define MCP_ADDR 0x67
Adafruit_MCP9601 mcp;
```

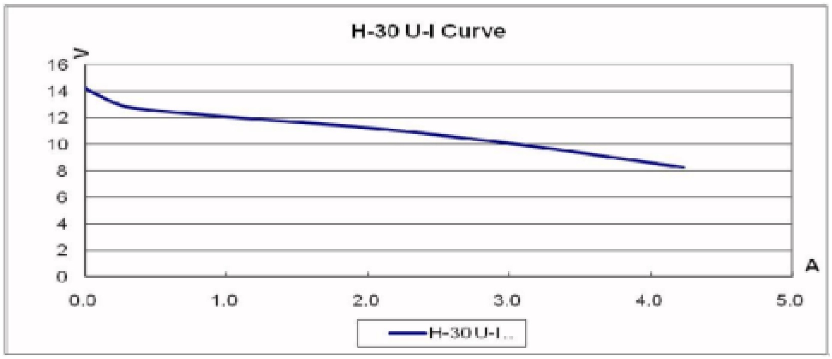
```
void setup() {
  Serial.begin(115200);
  Wire.begin();
  Wire.setClock(100000);

  Serial.println("🔧 Starting Fuel Cell Monitoring System...");

  // --- Initialize INA228s via TCA channel 0
  selectTCA(0);
  for (int i = 0; i < NUM_INA; i++) {
    Wire.beginTransmission(ina_addrs[i]);
    if (Wire.endTransmission() == 0) {
      writeINARegister16(ina_addrs[i], REG_SHUNT_CAL, 8192);
      Serial.print("✅ INA228 @ 0x");
      Serial.print(ina_addrs[i], HEX);
      Serial.println(" initialized.");
    } else {
      Serial.print("❌ INA228 @ 0x");
      Serial.print(ina_addrs[i], HEX);
      Serial.println(" not responding.");
    }
  }

  // --- Initialize MCP9601 on main I2C bus
  Serial.print("🔧 Initializing MCP9601 @ 0x");
  Serial.print(MCP_ADDR, HEX);
  Serial.println("...");
  if (!mcp.begin(MCP_ADDR)) {
    Serial.println("❌ MCP9601 not found. Check wiring!");
    while (1);
  }
  mcp.setThermocoupleType(MCP9600_TYPE_K);
  mcp.setAmbientResolution(RES_ZERO_POINT_0625);
  mcp.setADCresolution(MCP9600_ADCRESOLUTION_18);
  mcp.setFilterCoefficient(3);
  mcp.enable(true);
  Serial.println("✅ MCP9601 ready.");
}

void loop() {
  delay(5000);  // One reading every 5 seconds
  Serial.println("\n------ FUEL CELL SYSTEM SNAPSHOT ------");
```

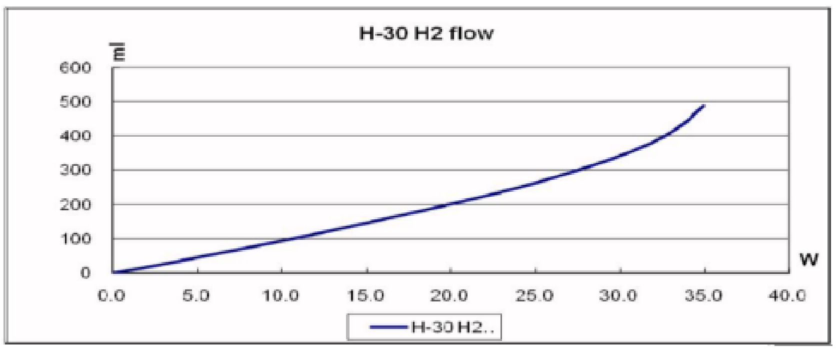
```
  // --- Read 3 INA228s via TCA
  selectTCA(0);
  for (int i = 0; i < NUM_INA; i++) {
    Serial.print("INA "); Serial.print(i + 1); Serial.print(" (0x");
    Serial.print(ina_addrs[i], HEX); Serial.println("):");

    int32_t rawV = readINARegister24(ina_addrs[i], REG_VBUS);
    float voltage = rawV * 0.1953125 / 1e6 * VBUS_SCALE;

    int32_t rawI = readINARegister24(ina_addrs[i], REG_CURRENT);
    float current = rawI * CURRENT_LSB;

    float power = voltage * current;

    Serial.print("  Voltage (V): "); Serial.println(voltage, 3);
    Serial.print("  Current (A): "); Serial.println(current, 3);
    Serial.print("  Power   (W): "); Serial.println(power, 3);
  }

  // --- Read MCP9601 temperature
  Serial.print("MCP9601 Temp (°C): ");
  float temp = mcp.readThermocouple();
  Serial.println(temp, 2);

  Serial.println("----------------------------------------");
}
```

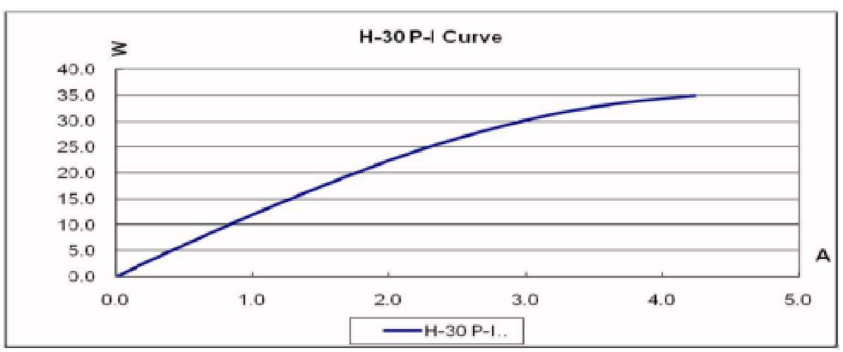
Appendix B:

Type of fuel cell	PEM
Number of cells	14
Rated Power	30W
Performance	8.4V @ 3.6A
Purging valve voltage	6V
Blower voltage	5V
Reactants	Hydrogen and Air
External temperature	5 to 30ºC
Max stack temperature	55ºC
H2 Pressure	0.45-0.55bar
Hydrogen purity	≧99.995% dry H2
Humidification	self-humidified
Cooling Weight (with fan & casing)	Air (integrated cooling fan) 280grams(±30 grams)
Controller	90 grams(±10 grams)
Dimension	8cmX4.7cmX7.5cm
Flow rate at max output*	0.42 L/min
Start up time	≦30S at ambient temperature
Efficiency of stack	40% @ at full power

* The flow rate may change with the power output.
** The Specification is subject to change without notice.







Appendix C:

Appendix D:

Appendix E: