

# **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 Contents	2
Table of Figures	3
Introduction:	3
Safety Precautions for Installation:	3
System Components:	4
Charging Station Components:	4
DAQ System Components:	5
Hydrogen Fuel Cells components:	7
System Design & Assembly:	11
Cart Design:	11
System Assembly:	12
DAQ system setup:	25
Sensor's setup and deployment:	25
Cloud Setup and Operation:	25
First Phase: Testing Connection to Azure with Mbed and DISCO-IOT101 Board	25
Second Phase: Sending the Environmental Sensor Telemetry to Azure(No Database Creation yet)	31
Third Phase: Storing NoSQL Data in Azure CosmosDB from Live Sensor Telemetry of DISCO-IOT Board	32
Operation and testing:	35
Charging the station:	35
Solar Panel Output:	37
Charge Controller Output:	39
Battery Voltage:	41
Discharging (Using a load):	42

Appendix A:	45
Appendix B:	49
Appendix C:	50
Appendix D:	50
Appendix E:	51

## Table of Figures

Figure 1: Cart structure .....	8
Figure 2: Lab Cart in closed position.....	9
Figure 3: Lab Cart in open position .....	9
Figure 4: Callouts for all the parts and materials .....	10
Figure 5: Solar panel hinge assembly with self-tapping screws .....	13
Figure 6: Inside hinge view under solar panel screws .....	14
Figure 7: Wooden panel support structure .....	15
Figure 8: Horizontal wood plank is slid in the groove and mounted to the solar panel using 1 screw on each side .....	16
Figure 9: The vertical wood plank is mounted with hinge and screws.....	17
Figure 10: The wood Structure is attached to the panel and rest on the inside cart edge fully extended to 45 degrees.....	18
Figure 11: Using 2 screws on each mount, mount the angle brackets .....	19
Figure 12: 3D Printed angle mounts are placed on 10 degree increments.....	20
Figure 13: Wood Pieces are placed to stop the vertical angle from moving side to side .....	21

## 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:

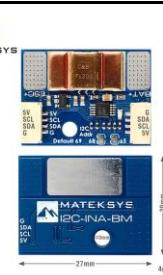
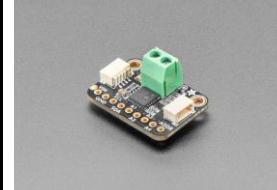
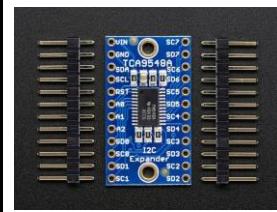
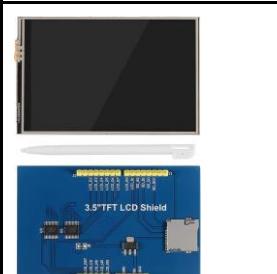
### Charging Station Components:

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.	

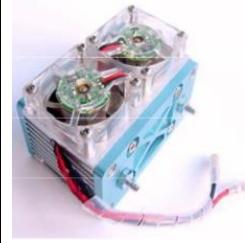
## DAQ System Components:

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 I <sup>2</sup> C 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 I <sup>2</sup> C 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.	

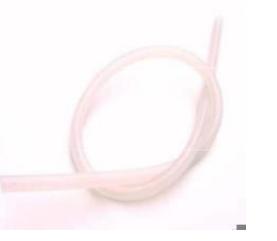
Purge valve	It controls the purging time for purging water and redundant Hydrogen from the fuel cells.	
Tube	It is used for connecting the pressure regulator to the stack, the stack to the purge valve and the purge valve to a place away from the stack.	



Figure 1: Cart structure

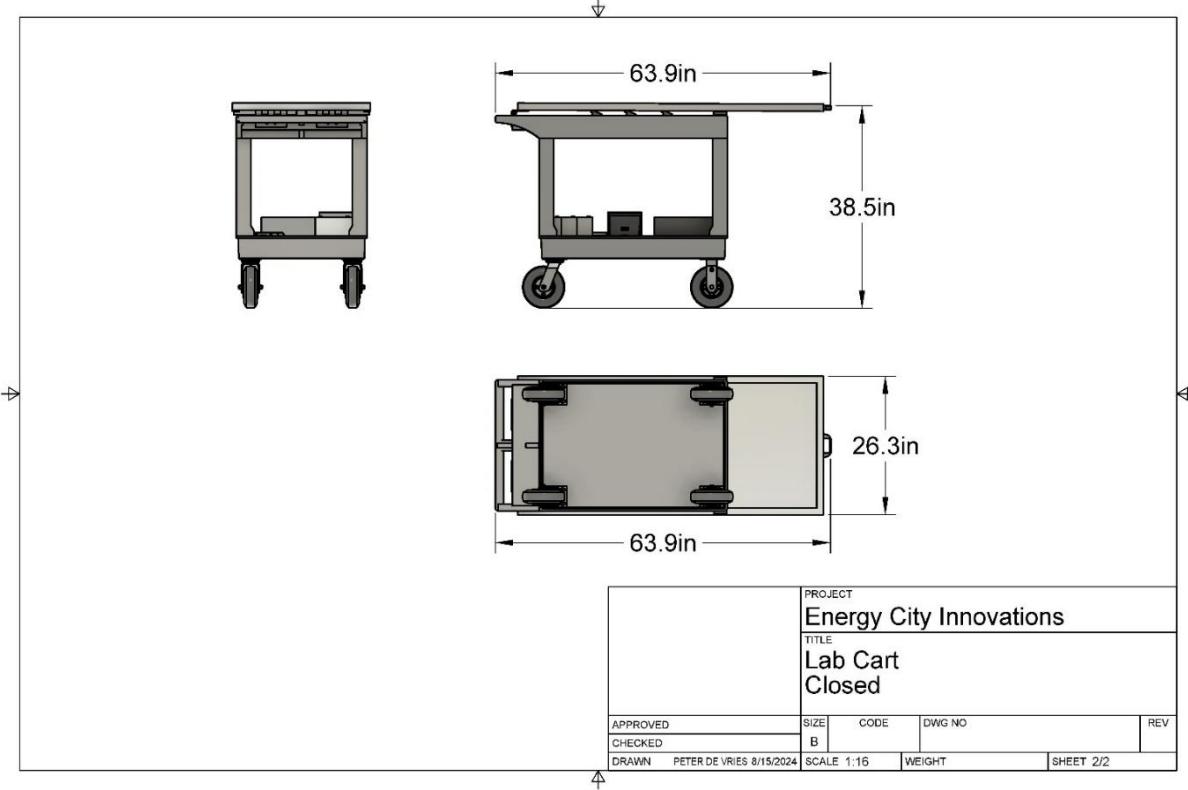


Figure 2: Lab Cart in closed position

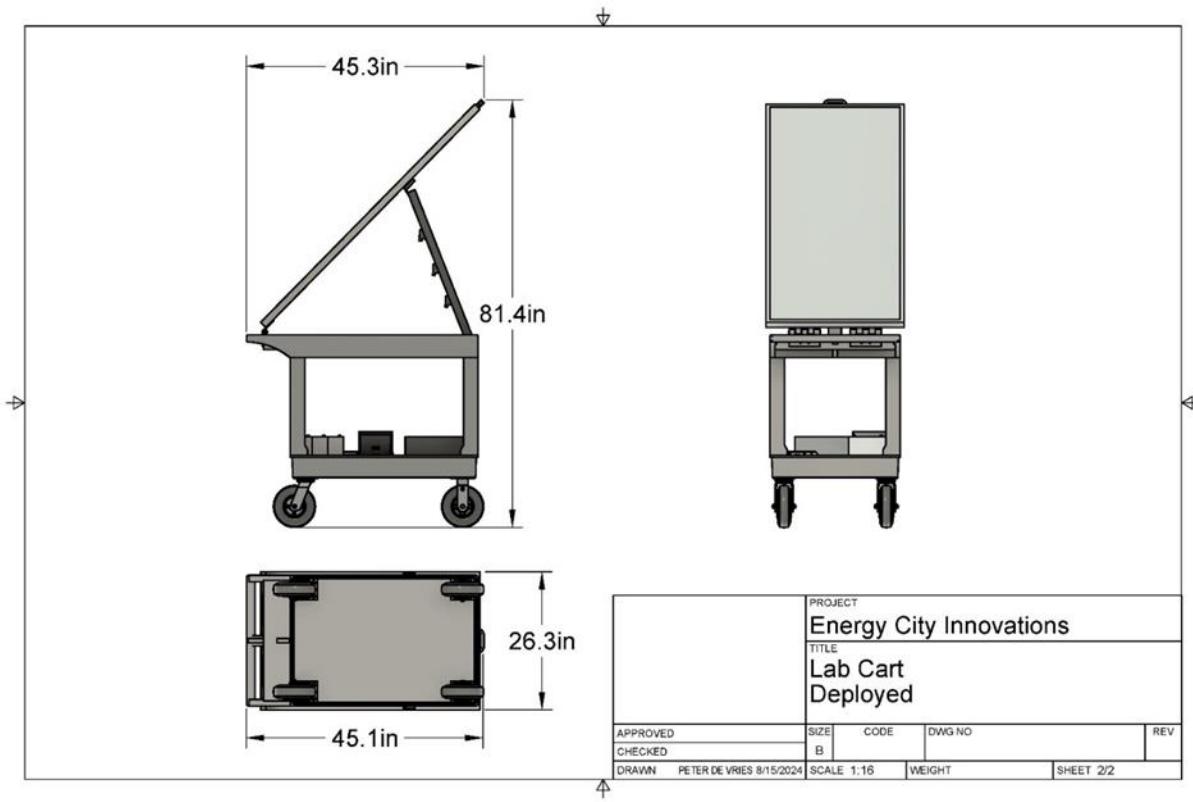


Figure 3: Lab Cart in open position

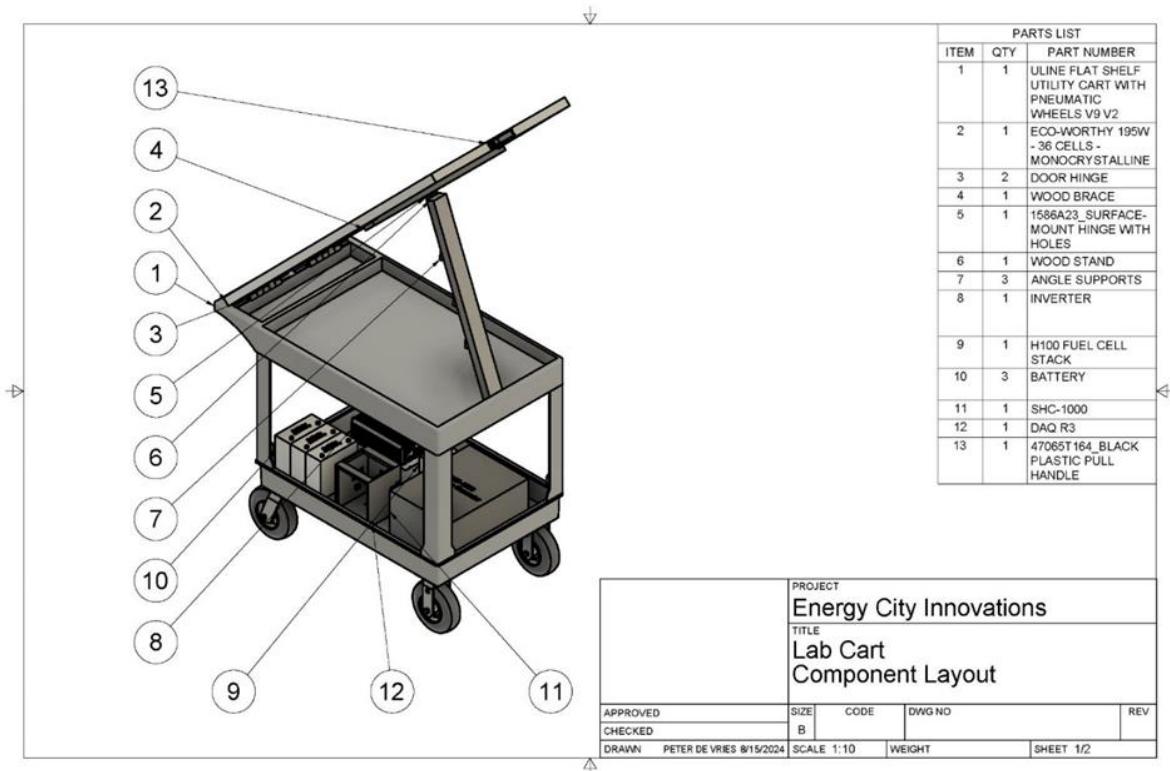
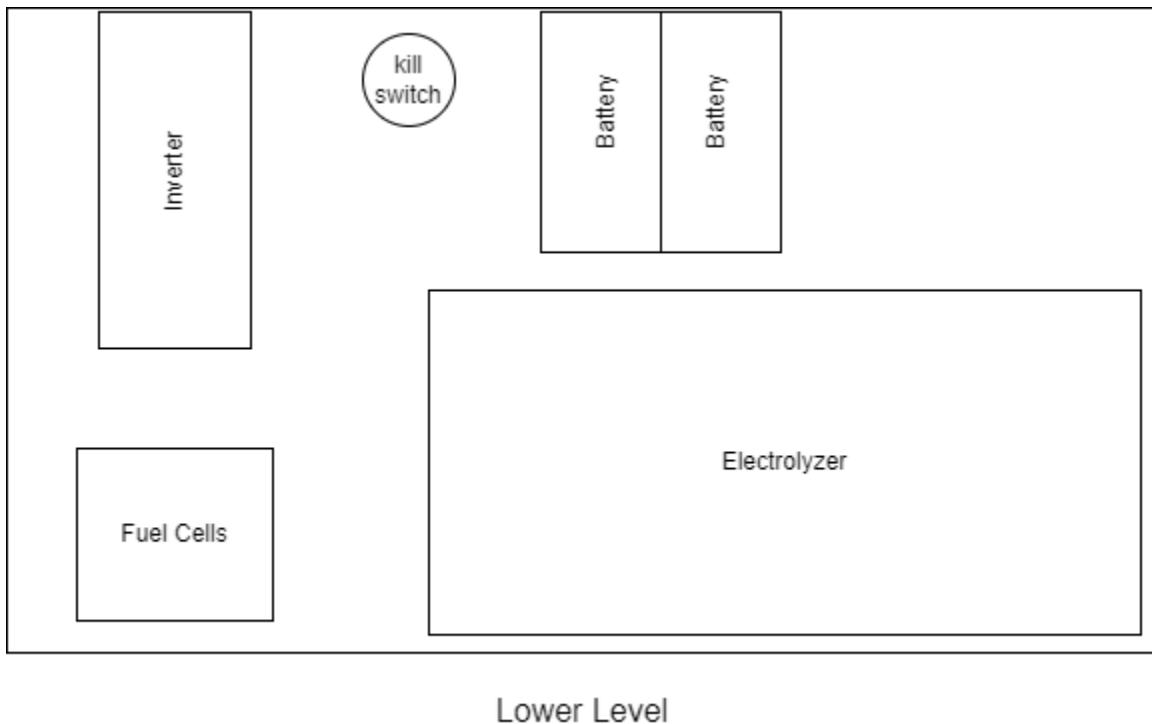


Figure 4: Callouts for all the parts and materials

# System Design & Assembly:

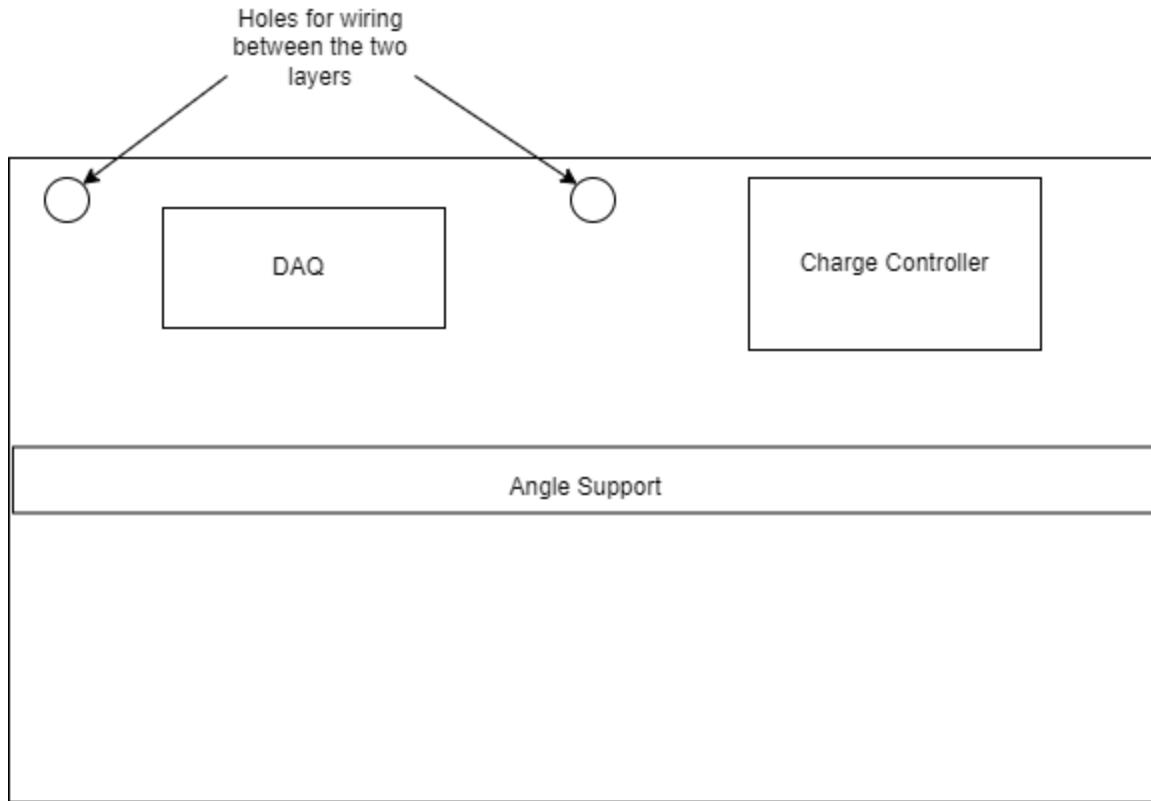
## System Layout:

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.



Lower Level

*Figure. Layout for the lower level of the cart*



**Upper-Level**

*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.

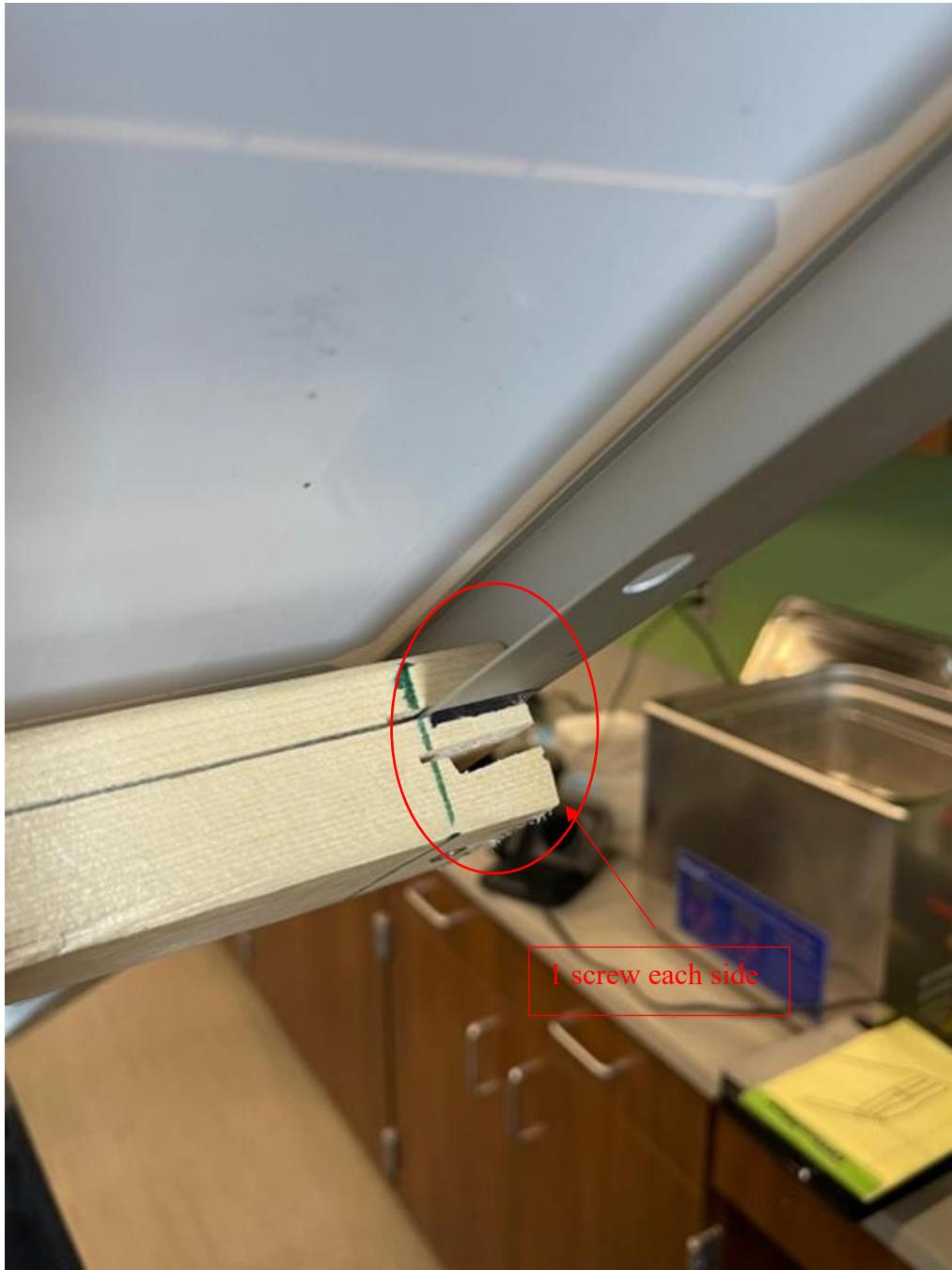


Figure 8: Horizontal wood plank is slid in the groove and mounted to the solar panel using 1 screw on each side

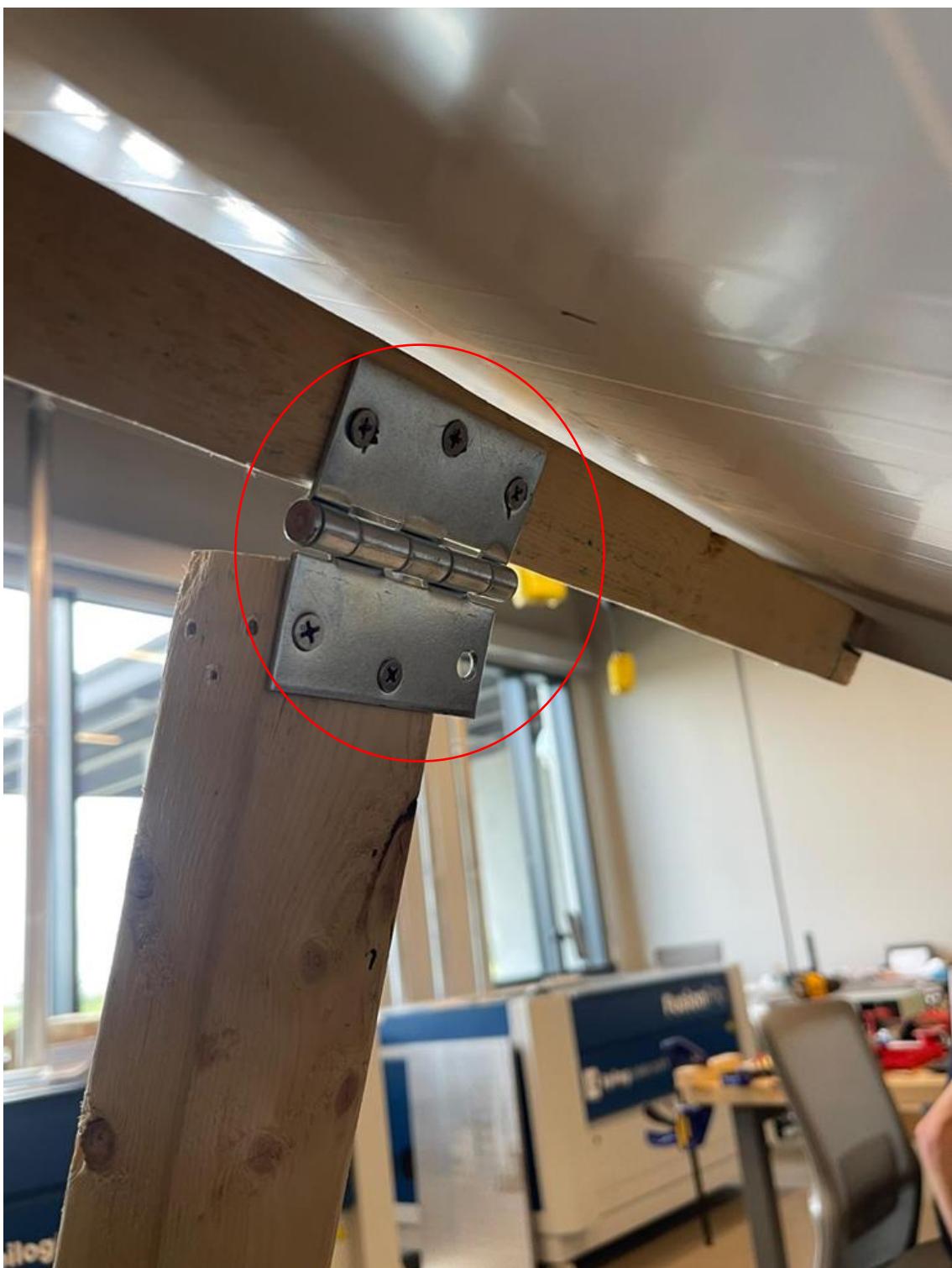


Figure 9: The vertical wood plank is mounted with hinge and screws.



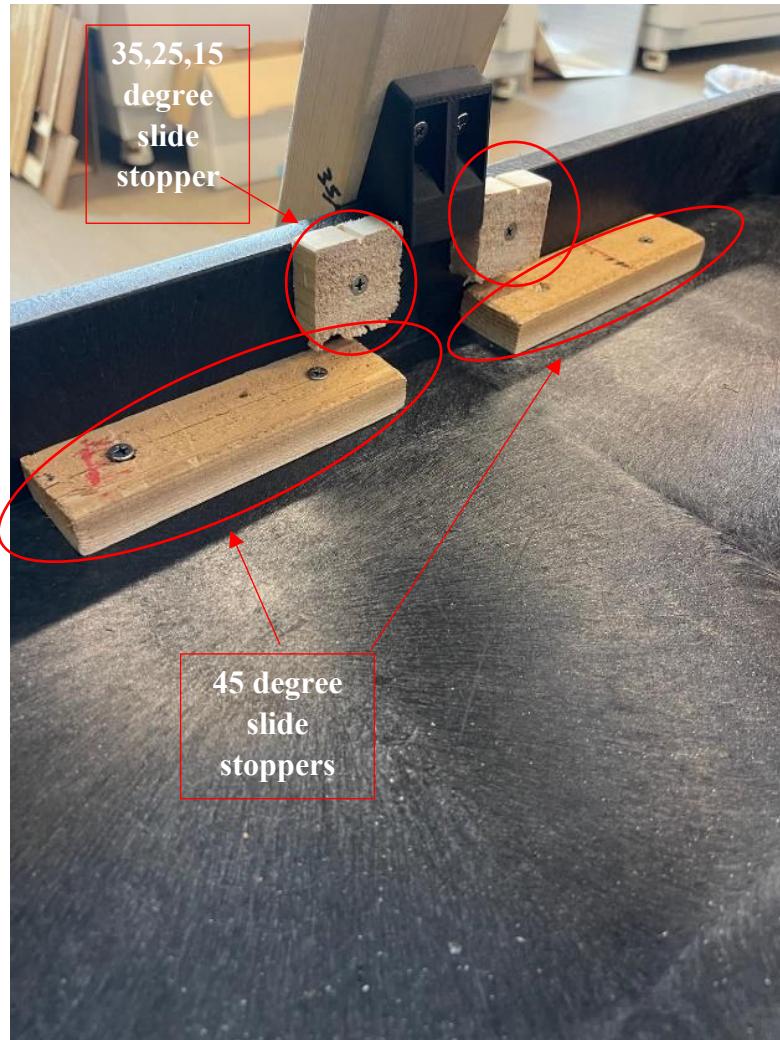
Figure 10: The wood Structure is attached to the panel and rest on the inside cart edge fully extended to 45 degrees.



*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  $\geq 99.995\%$  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.

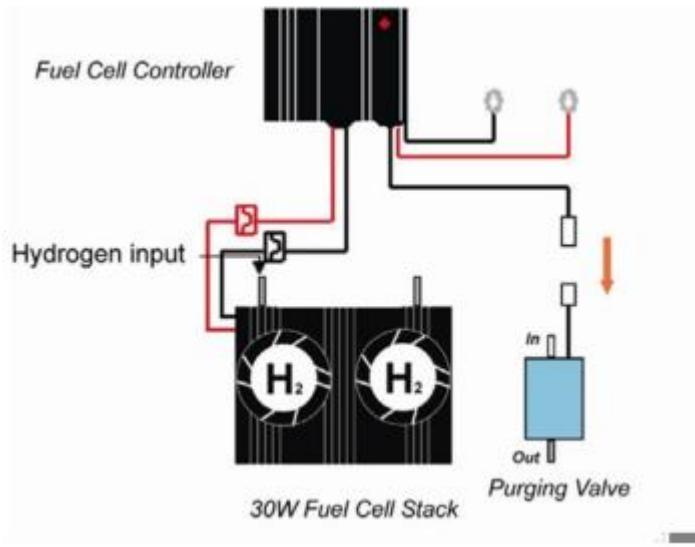


Figure. Dflsfsdp;asd,opkpoasg

**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.

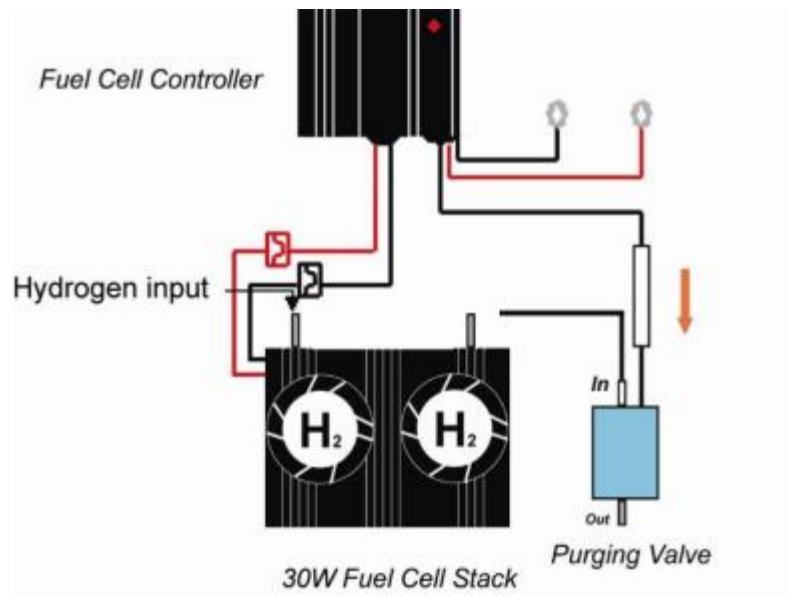


Figure. Dflp;asd,opkposasasg

**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.

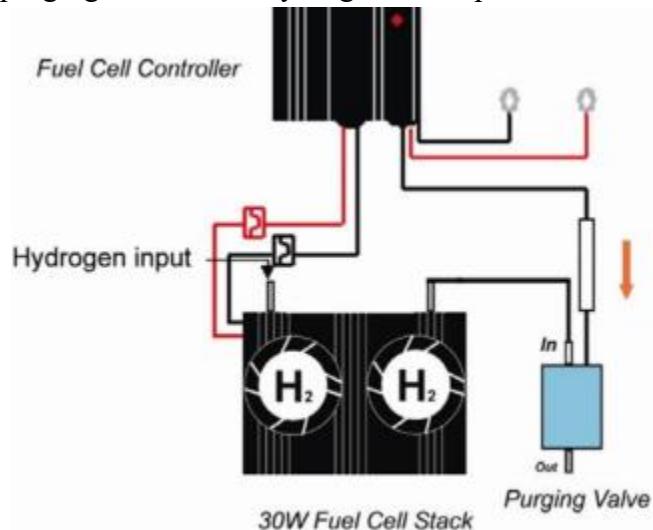


Figure. Dflp;asd,opkpoasasasdg

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

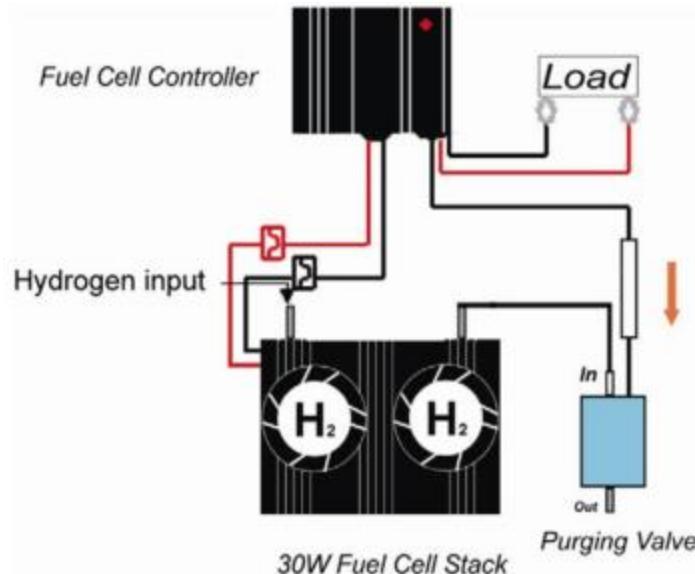
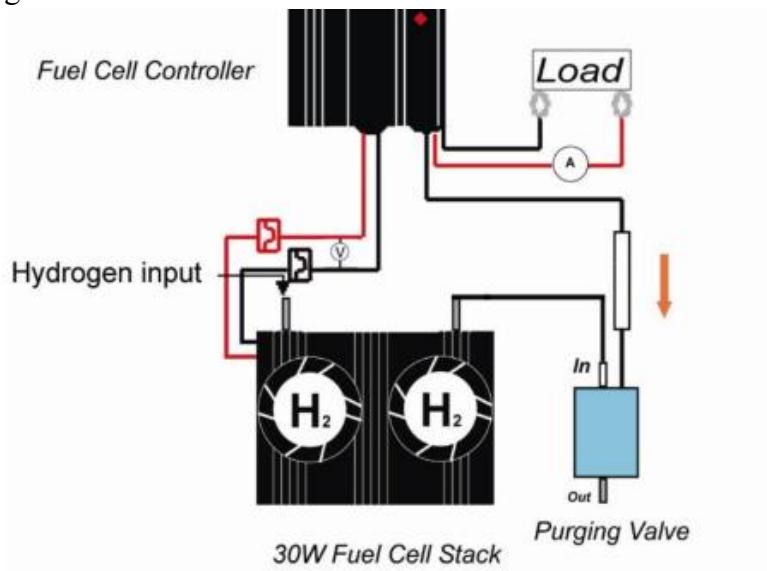


Figure. Dflp;asd,opkpoasgdasd

**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



$$\text{Power} = V \times A$$

Figure. Dflp;asd,adsfcasffaasdfs

#### *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:**

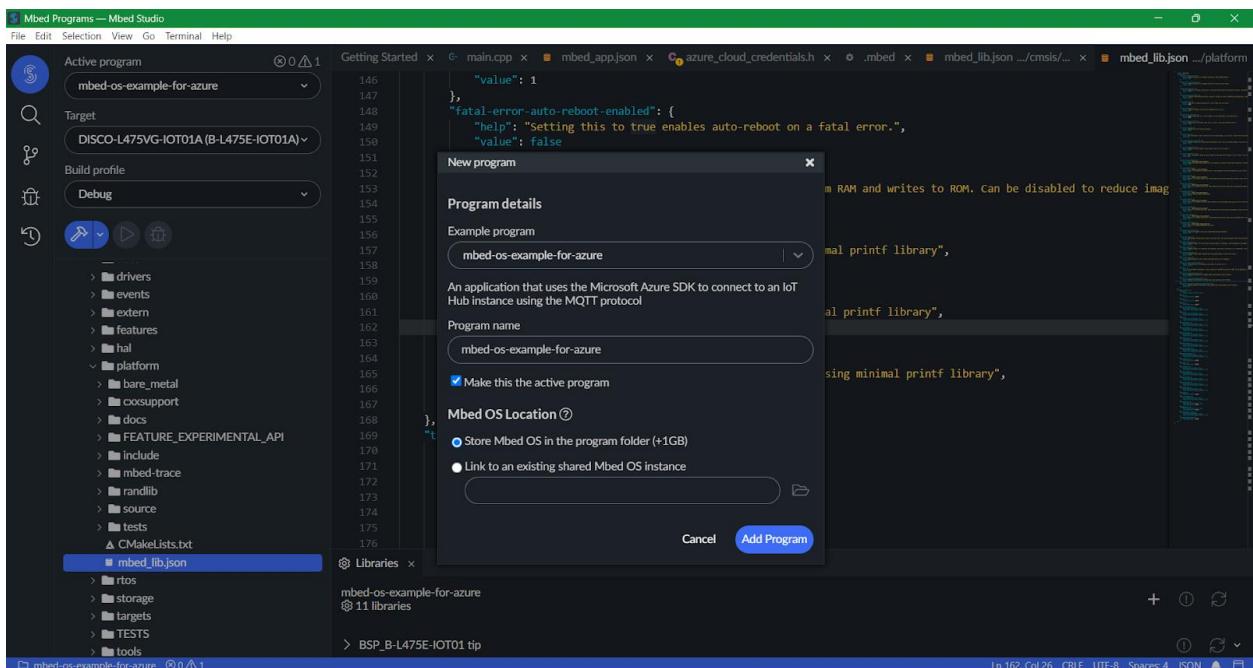
**Sensor's setup and deployment:**

## **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 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.

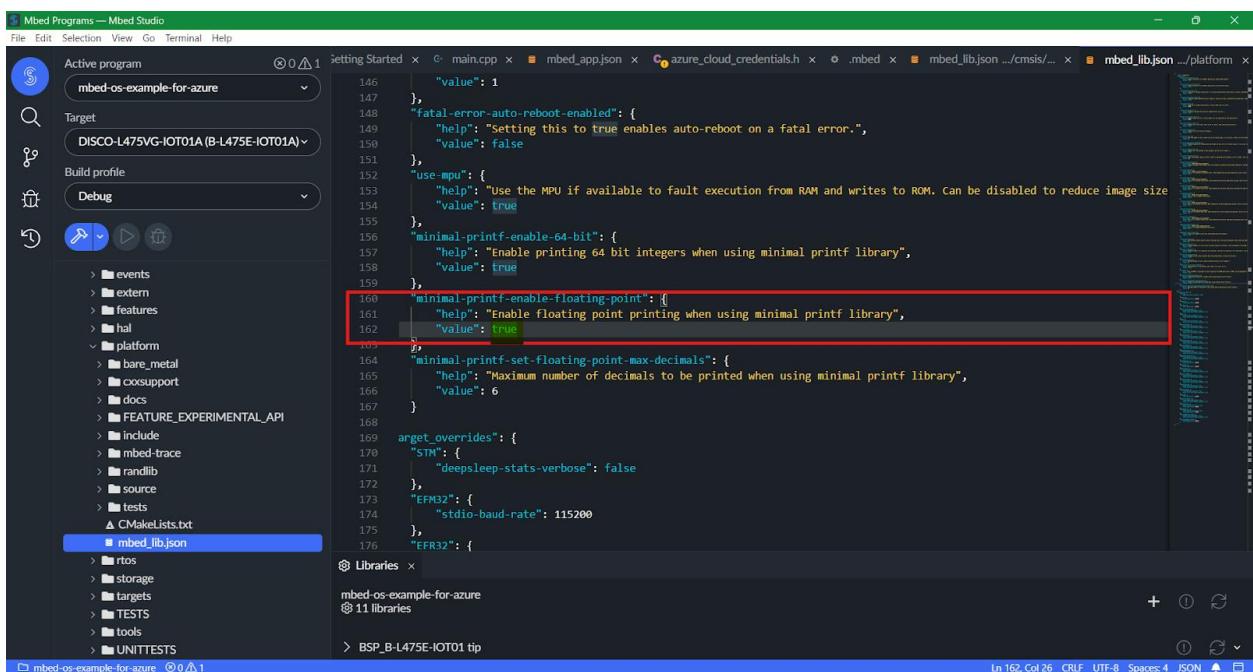


Figure. Enabling floating point printing in the mbed\_app.json file

- 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/](https://os.mbed.com/teams/ST/code/BSP_B-L475E-IOT01/) as a library to the active project:

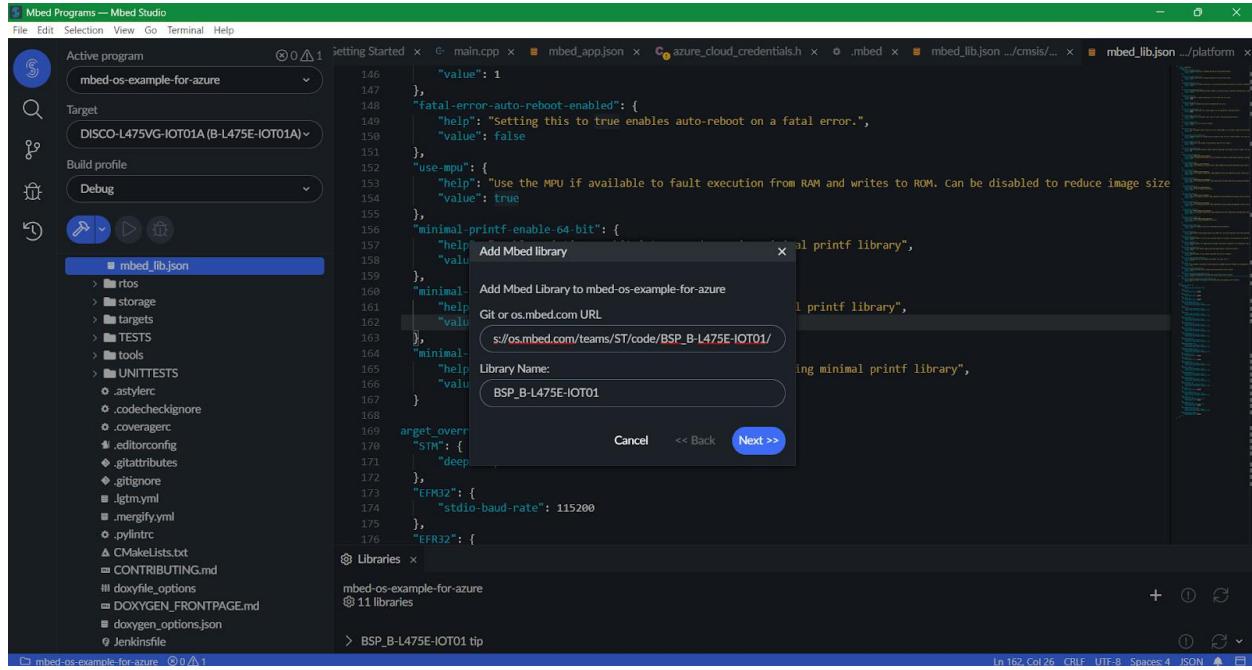


Figure n: Adding the BSP\_B-L475E-IOT01A library to the Mbed project using a Git URL

- 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.

The screenshot shows the Mbed Studio interface with the following details:

- Active program:** mbed-os-example-for-azure
- Target:** DISCO-L475VG-IOT01A (B-L475E-IOT01A)
- Build profile:** Debug
- File Explorer:** Shows files like main.cpp, mbed\_app.json, and azure\_cloud\_credentials.h.
- Code Editor:** Displays the mbed\_app.json file with the following configuration for the DISCO-L475VG\_IOT01A target:
 

```

    "target_overrides": {
      "DISCO_L475VG_IOT01A": {
        "target_components_add": ["wifi_ism43362"],
        "target_network-default-interface-type": "WIFI",
        "nsapi.default-wifi-security": "WPA_WPA2",
        "nsapi.default-wifi-ssid": "\\" + " " + "\\",
        "nsapi.default-wifi-password": "\\" + " " + "\\"
      }
    }
  
```
- Libraries:** Shows 11 libraries.

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.

The screenshot shows the Microsoft Azure portal interface with the following details:

- Header:** Microsoft Azure, Upgrade, Search resources, services, and docs, Copilot, DEFAULT DIRECTORY
- Azure services:** Shows icons for Create a resource, IoT Hub, Quickstart Center, Azure AI Foundry, Kubernetes services, Virtual machines, App Services, Storage accounts, SQL databases, and More services. The IoT Hub icon is highlighted with a red box and has a red arrow pointing to it from the left.
- Resources:** Shows a recent IoT Hub named arm-uttyler-hcc.
- Tools:** Subscriptions, Resource groups, All resources, Dashboard, Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management.

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

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

Home > IoT Hub ...

**IoT Hub**

Default Directory X

+ Create Manage view Refresh Export to CSV Open query Assign tags Group by none

You are viewing a new version of Browse experience. Click here to access the old experience.

Filter for any field... Subscription equals all Resource Group equals all Location equals all Add filter

Name ↑	Type	Resource Group	Location	Subscription
arm-uttyler-hcc	IoT Hub	arm-stm-iot	East US	Azure subscription 1

Newly created IoT Hub

Showing 1 - 1 of 1. Display count: auto

Give feedback

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

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

Microsoft Azure Upgrade

Search resources, services, and docs (G+) Copilot

Eng.AhmedMohd1@outlook.com DEFAULT DIRECTORY

Home > IoT Hub > arm-uttyler-hcc

**IoT Hub**

Default Directory

+ Create Manage view ...

You are viewing a new version of Browse experience. Click here to access the old experience.

Name ↑

arm-uttyler-hcc

**arm-uttyler-hcc | Devices**

IoT Hub

devices

View, create, delete, and update devices in your IoT Hub. Learn more

Add Device Edit columns Refresh Assign tags Delete

Find devices using a query

Device management

Devices

IoT Edge

Queries

Security settings

Shared access policies

Device ID

Type

Status

Last status update

Auth...

C2D ...

Tags

DISCO-id1

IoT Edge Device

Enabled

--

Shared... 0

Newly created device for our DISCO board

Showing 1 - 1 of 1. Display count: auto

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.

Microsoft Azure Upgrade Search resources, services, and docs (G+)

Home > IoT Hub > arm-uttyler-hcc | Devices > DISCO-id1

DISCO-id1 ...

arm-uttyler-hcc

Save Manage keys Set modules Manage child devices Troubleshoot Device twin Refresh

Device ID: DISCO-id1

Primary key: [REDACTED]

Secondary key: [REDACTED]

Primary connection string: [REDACTED]

Secondary connection string: [REDACTED]

IoT Edge runtime response: NA

Tags (edit): No tags

Enable connection to IoT Hub:  Enable  Disable

Parent device: No parent device

Modules IoT Edge hub connections Deployments and Configurations

Name	Type	Specified in Deployment	Reported by Device	Runtime Status	Exit Code
\$edgeAgent	Module Identity	NA	NA	NA	NA
\$edgeHub	Module Identity	NA	NA	NA	NA

This goes into azure\_cloud\_credentials.h

*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.

The screenshot shows the Mbed Studio interface. On the left, the file tree displays the project structure for 'mbed-os-example-for-azure' targeting 'DISCO-L475VG-IOT01A (B-L475E-IOT...)'. In the center, the code editor shows the 'main.cpp' file with the following content:

```

1  /*
2   * Copyright (c) 2020 Arm Limited
3   * SPDX-License-Identifier: Apache-2.0
4   */
5
6 }
7
8 {
9     "event": {
10         "origin": "DISCO-id1",
11         "module": "",
12         "interface": "",
13         "component": "",
14         "payload": "3 messages left to send, or until we receive a reply"
15     }
16
17     "event": {
18         "origin": "DISCO-id1",
19         "module": "",
20         "interface": "",
21         "component": "",
22         "payload": "2 messages left to send, or until we receive a reply"
23     }
24
25     "event": {
26         "origin": "DISCO-id1",
27         "module": "",
28         "interface": "",
29         "component": "",
30         "payload": "1 messages left to send, or until we receive a reply"
31     }
32
33 }
34
35 }
36
37 }
38
39 }
40
41 }
42
43 }
44
45 }
46
47 }
48
49 }
50
51 }
52
53 }
54
55 }
56
57 }
58
59 }
60
61 }
62
63 }
64
65 }
66
67 }
68
69 }
70
71 }
72
73 }
74
75 }
76
77 }
78
79 }
80
81 }
82
83 }
84
85 }
86
87 }
88
89 }
90
91 }
92
93 }
94
95 }
96
97 }
98
99 }
100
101 }
102
103 }
104
105 }
106
107 }
108
109 }
110
111 }
112
113 }
114
115 }
116
117 }
118
119 }
120
121 }
122
123 }
124
125 }
126
127 }
128
129 }
130
131 }
132
133 }
134
135 }
136
137 }
138
139 }
140
141 }
142
143 }
144
145 }
146
147 }
148
149 }
150
151 }
152
153 }
154
155 }
156
157 }
158
159 }
160
161 }
162
163 }
164
165 }
166
167 }
168
169 }
170
171 }
172
173 }
174
175 }
176
177 }
178
179 }
180
181 }
182
183 }
184
185 }
186
187 }
188
189 }
190
191 }
192
193 }
194
195 }
196
197 }
198
199 }
200
201 }
202
203 }
204
205 }
206
207 }
208
209 }
210
211 }
212
213 }
214
215 }
216
217 }
218
219 }
220
221 }
222
223 }
224
225 }
226
227 }
228
229 }
230
231 }
232
233 }
234
235 }
236
237 }
238
239 }
240
241 }
242
243 }
244
245 }
246
247 }
248
249 }
250
251 }
252
253 }
254
255 }
256
257 }
258
259 }
259
260 }
261
262 }
263
264 }
265
266 }
267
268 }
269
270 }
271
272 }
273
274 }
275
276 }
277
278 }
279
280 }
281
282 }
283
284 }
285
286 }
287
288 }
289
290 }
291
292 }
293
294 }
295
296 }
297
298 }
299
299
300 }
301
302 }
303
304 }
305
306 }
307
308 }
309
310 }
311
312 }
313
314 }
315
316 }
317
318 }
319
320 }
321
322 }
323
324 }
325
326 }
327
328 }
329
330 }
331
332 }
333
334 }
335
336 }
337
338 }
339
340 }
341
342 }
343
344 }
345
346 }
347
348 }
349
349
350 }
351
352 }
353
354 }
355
356 }
357
358 }
359
359
360 }
361
362 }
363
364 }
365
366 }
367
368 }
369
369
370 }
371
372 }
373
374 }
375
376 }
377
378 }
379
379
380 }
381
382 }
383
384 }
385
386 }
387
388 }
389
389
390 }
391
392 }
393
394 }
395
396 }
397
398 }
399
399
400 }
401
402 }
403
404 }
405
406 }
407
408 }
409
409
410 }
411
412 }
413
414 }
415
416 }
417
418 }
419
419
420 }
421
422 }
423
424 }
425
426 }
427
428 }
429
429
430 }
431
432 }
433
434 }
435
436 }
437
438 }
439
439
440 }
441
442 }
443
444 }
445
446 }
447
448 }
449
449
450 }
451
452 }
453
454 }
455
456 }
457
458 }
459
459
460 }
461
462 }
463
464 }
465
466 }
467
468 }
469
469
470 }
471
472 }
473
474 }
475
476 }
477
478 }
479
479
480 }
481
482 }
483
484 }
485
486 }
487
488 }
489
489
490 }
491
492 }
493
494 }
495
496 }
497
498 }
499
499
500 }
501
502 }
503
504 }
505
506 }
507
508 }
509
509
510 }
511
512 }
513
514 }
515
516 }
517
518 }
519
519
520 }
521
522 }
523
524 }
525
526 }
527
528 }
529
529
530 }
531
532 }
533
534 }
535
536 }
537
538 }
539
539
540 }
541
542 }
543
544 }
545
546 }
547
548 }
549
549
550 }
551
552 }
553
554 }
555
556 }
557
558 }
559
559
560 }
561
562 }
563
564 }
565
566 }
567
568 }
569
569
570 }
571
572 }
573
574 }
575
576 }
577
578 }
579
579
580 }
581
582 }
583
584 }
585
586 }
587
588 }
589
589
590 }
591
592 }
593
594 }
595
596 }
597
598 }
599
599
600 }
601
602 }
603
604 }
605
606 }
607
608 }
609
609
610 }
611
612 }
613
614 }
615
616 }
617
618 }
619
619
620 }
621
622 }
623
624 }
625
626 }
627
628 }
629
629
630 }
631
632 }
633
634 }
635
636 }
637
638 }
639
639
640 }
641
642 }
643
644 }
645
646 }
647
648 }
649
649
650 }
651
652 }
653
654 }
655
656 }
657
658 }
659
659
660 }
661
662 }
663
664 }
665
666 }
667
668 }
669
669
670 }
671
672 }
673
674 }
675
676 }
677
678 }
679
679
680 }
681
682 }
683
684 }
685
686 }
687
688 }
689
689
690 }
691
692 }
693
694 }
695
696 }
697
698 }
699
699
700 }
701
702 }
703
704 }
705
706 }
707
708 }
709
709
710 }
711
712 }
713
714 }
715
716 }
717
718 }
719
719
720 }
721
722 }
723
724 }
725
726 }
727
728 }
729
729
730 }
731
732 }
733
734 }
735
736 }
737
738 }
739
739
740 }
741
742 }
743
744 }
745
746 }
747
748 }
749
749
750 }
751
752 }
753
754 }
755
756 }
757
758 }
759
759
760 }
761
762 }
763
764 }
765
766 }
767
768 }
769
769
770 }
771
772 }
773
774 }
775
776 }
777
778 }
779
779
780 }
781
782 }
783
784 }
785
786 }
787
788 }
789
789
790 }
791
792 }
793
794 }
795
796 }
797
798 }
799
799
800 }
801
802 }
803
804 }
805
806 }
807
808 }
809
809
810 }
811
812 }
813
814 }
815
816 }
817
818 }
819
819
820 }
821
822 }
823
824 }
825
826 }
827
828 }
829
829
830 }
831
832 }
833
834 }
835
836 }
837
838 }
839
839
840 }
841
842 }
843
844 }
845
846 }
847
848 }
849
849
850 }
851
852 }
853
854 }
855
856 }
857
858 }
859
859
860 }
861
862 }
863
864 }
865
866 }
867
868 }
869
869
870 }
871
872 }
873
874 }
875
876 }
877
878 }
879
879
880 }
881
882 }
883
884 }
885
886 }
887
888 }
889
889
890 }
891
892 }
893
894 }
895
896 }
897
898 }
899
899
900 }
901
902 }
903
904 }
905
906 }
907
908 }
909
909
910 }
911
912 }
913
914 }
915
916 }
917
918 }
919
919
920 }
921
922 }
923
924 }
925
926 }
927
928 }
929
929
930 }
931
932 }
933
934 }
935
936 }
937
938 }
939
939
940 }
941
942 }
943
944 }
945
946 }
947
948 }
949
949
950 }
951
952 }
953
954 }
955
956 }
957
958 }
959
959
960 }
961
962 }
963
964 }
965
966 }
967
968 }
969
969
970 }
971
972 }
973
974 }
975
976 }
977
978 }
979
979
980 }
981
982 }
983
984 }
985
986 }
987
988 }
989
989
990 }
991
992 }
993
994 }
995
996 }
997
998 }
999
999
1000 }
1001
1002 }
1003
1004 }
1005
1006 }
1007
1008 }
1009
1009
1010 }
1011
1012 }
1013
1014 }
1015
1016 }
1017
1018 }
1019
1019
1020 }
1021
1022 }
1023
1024 }
1025
1026 }
1027
1028 }
1029
1029
1030 }
1031
1032 }
1033
1034 }
1035
1036 }
1037
1038 }
1039
1039
1040 }
1041
1042 }
1043
1044 }
1045
1046 }
1047
1048 }
1049
1049
1050 }
1051
1052 }
1053
1054 }
1055
1056 }
1057
1058 }
1059
1059
1060 }
1061
1062 }
1063
1064 }
1065
1066 }
1067
1068 }
1069
1069
1070 }
1071
1072 }
1073
1074 }
1075
1076 }
1077
1078 }
1079
1079
1080 }
1081
1082 }
1083
1084 }
1085
1086 }
1087
1088 }
1089
1089
1090 }
1091
1092 }
1093
1094 }
1095
1096 }
1097
1098 }
1099
1099
1100 }
1101
1102 }
1103
1104 }
1105
1106 }
1107
1108 }
1109
1109
1110 }
1111
1112 }
1113
1114 }
1115
1116 }
1117
1118 }
1119
1119
1120 }
1121
1122 }
1123
1124 }
1125
1126 }
1127
1128 }
1129
1129
1130 }
1131
1132 }
1133
1134 }
1135
1136 }
1137
1138 }
1139
1139
1140 }
1141
1142 }
1143
1144 }
1145
1146 }
1147
1148 }
1149
1149
1150 }
1151
1152 }
1153
1154 }
1155
1156 }
1157
1158 }
1159
1159
1160 }
1161
1162 }
1163
1164 }
1165
1166 }
1167
1168 }
1169
1169
1170 }
1171
1172 }
1173
1174 }
1175
1176 }
1177
1178 }
1179
1179
1180 }
1181
1182 }
1183
1184 }
1185
1186 }
1187
1188 }
1189
1189
1190 }
1191
1192 }
1193
1194 }
1195
1196 }
1197
1198 }
1199
1199
1200 }
1201
1202 }
1203
1204 }
1205
1206 }
1207
1208 }
1209
1209
1210 }
1211
1212 }
1213
1214 }
1215
1216 }
1217
1218 }
1219
1219
1220 }
1221
1222 }
1223
1224 }
1225
1226 }
1227
1228 }
1229
1229
1230 }
1231
1232 }
1233
1234 }
1235
1236 }
1237
1238 }
1239
1239
1240 }
1241
1242 }
1243
1244 }
1245
1246 }
1247
1248 }
1249
1249
1250 }
1251
1252 }
1253
1254 }
1255
1256 }
1257
1258 }
1259
1259
1260 }
1261
1262 }
1263
1264 }
1265
1266 }
1267
1268 }
1269
1269
1270 }
1271
1272 }
1273
1274 }
1275
1276 }
1277
1278 }
1279
1279
1280 }
1281
1282 }
1283
1284 }
1285
1286 }
1287
1288 }
1289
1289
1290 }
1291
1292 }
1293
1294 }
1295
1296 }
1297
1298 }
1299
1299
1300 }
1301
1302 }
1303
1304 }
1305
1306 }
1307
1308 }
1309
1309
1310 }
1311
1312 }
1313
1314 }
1315
1316 }
1317
1318 }
1319
1319
1320 }
1321
1322 }
1323
1324 }
1325
1326 }
1327
1328 }
1329
1329
1330 }
1331
1332 }
1333
1334 }
1335
1336 }
1337
1338 }
1339
1339
1340 }
1341
1342 }
1343
1344 }
1345
1346 }
1347
1348 }
1349
1349
1350 }
1351
1352 }
1353
1354 }
1355
1356 }
1357
1358 }
1359
1359
1360 }
1361
1362 }
1363
1364 }
1365
1366 }
1367
1368 }
1369
1369
1370 }
1371
1372 }
1373
1374 }
1375
1376 }
1377
1378 }
1379
1379
1380 }
1381
1382 }
1383
1384 }
1385
1386 }
1387
1388 }
1389
1389
1390 }
1391
1392 }
1393
1394 }
1395
1396 }
1397
1398 }
1399
1399
1400 }
1401
1402 }
1403
1404 }
1405
1406 }
1407
1408 }
1409
1409
1410 }
1411
1412 }
1413
1414 }
1415
1416 }
1417
1418 }
1419
1419
1420 }
1421
1422 }
1423
1424 }
1425
1426 }
1427
1428 }
1429
1429
1430 }
1431
1432 }
1433
1434 }
1435
1436 }
1437
1438 }
1439
1439
1440 }
1441
1442 }
1443
1444 }
1445
1446 }
1447
1448 }
1449
1449
1450 }
1451
1452 }
1453
1454 }
1455
1456 }
1457
1458 }
1459
1459
1460 }
1461
1462 }
1463
1464 }
1465
1466 }
1467
1468 }
1469
1469
1470 }
1471
1472 }
1473
1474 }
1475
1476 }
1477
1478 }
1479
1479
1480 }
1481
1482 }
1483
1484 }
1485
1486 }
1487
1488 }
1489
1489
1490 }
1491
1492 }
1493
1494 }
1495
1496 }
1497
1498 }
1499
1499
1500 }
1501
1502 }
1503
1504 }
1505
1506 }
1507
1508 }
1509
1509
1510 }
1511
1512 }
1513
1514 }
1515
1516 }
1517
1518 }
1519
1519
1520 }
1521
1522 }
1523
1524 }
1525
1526 }
1527
1528 }
1529
1529
1530 }
1531
1532 }
1533
1534 }
1535
1536 }
1537
1538 }
1539
1539
1540 }
1541
1542 }
1543
1544 }
1545
1546 }
1547
1548 }
1549
1549
1550 }
1551
1552 }
1553
1554 }
1555
1556 }
1557
1558 }
1559
1559
1560 }
1561
1562 }
1563
1564 }
1565
1566 }
1567
1568 }
1569
1569
1570 }
1571
1572 }
1573
1574 }
1575
1576 }
1577
1578 }
1579
1579
1580 }
1581
1582 }
1583
1584 }
1585
1586 }
1587
1588 }
1589
1589
1590 }
1591
1592 }
1593
1594 }
1595
1596 }
1597
1598 }
1599
1599
1600 }
1601
1602 }
1603
1604 }
1605
1606 }
1607
1608 }
1609
1609
1610 }
1611
1612 }
1613
1614 }
1615
1616 }
1617
1618 }
1619
1619
1620 }
1621
1622 }
1623
1624 }
1625
1626 }
1627
1628 }
1629
1629
1630 }
1631
1632 }
1633
1634 }
1635
1636 }
1637
1638 }
1639
1639
1640 }
1641
1642 }
1643
1644 }
1645
1646 }
1647
1648 }
1649
1649
1650 }
1651
1652 }
1653
1654 }
1655
1656 }
1657
1658 }
1659
1659
1660 }
1661
1662 }
1663
1664 }
1665
1666 }
1667
1668 }
1669
1669
1670 }
1671
1672 }
1673
1674 }
1675
1676 }
1677
1678 }
1679
1679
1680 }
1681
1682 }
1683
1684 }
1685
1686 }
1687
1688 }
1689
1689
1690 }
1691
1692 }
1693
1694 }
1695
1696 }
1697
1698 }
1699
1699
1700 }
1701
1702 }
1703
1704 }
1705
1706 }
1707
1708 }
1709
1709
1710 }
1711
1712 }
1713
1714 }
1715
1716 }
1717
1718 }
1719
1719
1720 }
1721
1722 }
1723
1724 }
1725
1726 }
1727
1728 }
1729
1729
1730 }
1731
1732 }
1733
1734 }
1735
1736 }
1737
1738 }
1739
1739
1740 }
1741
1742 }
1743
1744 }
1745
1746 }
1747
1748 }
1749
1749
1750 }
1751
1752 }
1753
1754 }
1755
1756 }
1757
1758 }
1759
1759
1760 }
1761
1762 }
1763
1764 }
1765
1766 }
1767
1768 }
1769
1769
1770 }
1771
1772 }
1773
1774 }
1775
1776 }
1777
1778 }
1779
1779
1780 }
1781
1782 }
1783
1784 }
1785
1786 }
1787
1788 }
1789
1789
1790 }
1791
1792 }
1793
1794 }
1795
1796 }
1797
1798 }
1799
1799
1800 }
1801
1802 }
1803
1804 }
1805
1806 }
1807
1808 }
1809
1809
1810 }
1811
1812 }
1813
1814 }
1815
1816 }
1817
1818 }
1819
1819
1820 }
1821
1822 }
1823
1824 }
1825
1826 }
1827
1828 }
1829
1829
1830 }
1831
1832 }
1833
1834 }
1835
1836 }
1837
1838 }
1839
1839
1840 }
1841
1842 }
1843
1844 }
1845
1846 }
1847
1848 }
1849
1849
1850 }
1851
1852 }
1853
1854 }
1855
1856 }
1857
1858 }
1859
1859
1860 }
1861
1862 }
1863
1864 }
1865
1866 }
1867
1868 }
1869
1869
1870 }
1871
1872 }
1873
1874 }
1875
1876 }
1877
1878 }
1879
1879
1880 }
1881
1882 }
1883
1884 }
1885
1886 }
1887
1888 }
1889
1889
1890 }
1891
1892 }
1893
1894 }
1895
1896 }
1897
1898 }
1899
1899
1900 }
1901
1902 }
1903
1904 }
1905
1906 }
1907
1908 }
1909
1909
1910 }
1911
1912 }
1913
1914 }
1915
1916 }
1917
1918 }
1919
1919
1920 }
1921
1922 }
1923
1924 }
1925
1926 }
1927
1928 }
1929
1929
1930 }
1931
1932 }
1933
1934 }
1935
1936 }
1937
1938 }
1939
1939
1940 }
1941
1942 }
1943
1944 }
1945
1946 }
1947
1948 }
1949
1949
1950 }
1951
1952 }
1953
1954 }
1955
1956 }
1957
1958 }
1959
1959
1960 }
1961
1962 }
1963
1964 }
1965
1966 }
1967
1968 }
1969
1969
1970 }
1971
1972 }
1973
1974 }
1975
1976 }
1977
1978 }
1979
1979
1980 }
1981
1982 }
1983
1984 }
1985
1986 }
1987
1988 }
1989
1989
1990 }
1991
1992 }
1993
1994 }
1995
1996 }
1997
1998 }
1999
1999
2000 }
2001
2002 }
2003
2004 }
2005
2006 }
2007
2008 }
2009
2009
2010 }
2011
2012 }
2013
2014 }
2015
2016 }
2017
2018 }
2019
2019
2020 }
2021
2022 }
2023
2024 }
2025
2026 }
2027
2028 }
2029
2029
2030 }
2031
2032 }
2033
2034 }
2035
2036 }
2037
2038 }
2039
2039
2040 }
2041
2042 }
2043
2044 }
2045
2046 }
2047
2048 }
2049
2049
2050 }
2051
2052 }
2053
2054 }
2055
2056 }
2057
2058 }
2059
2059
2060 }
2061
2062 }
2063
2064 }
2065
2066 }
2067
2068 }
2069
2069
2070 }
2071
2072 }
2073
2074 }
2075
2076 }
2077
2078 }
2079
2079
2080 }
2081
2082 }
2083
2084 }
2085
2086 }
2087
2088 }
2089
2089
2090 }
2091
2092 }
2093
2094 }
2095
2096 }
2097
2098 }
2099
2099
2100 }
2101
2102 }
2103
2104 }
2105
2106 }
2107
2108 }
2109
2109
2110 }
2111
2112 }
2113
2114 }
2115
2116 }
2117
2118 }
2119
2119
2120 }
2121
2122 }
2123
2124 }
2125
2126 }
2127
2128 }
2129
2129
2130 }
2131
2132 }
2133
2134 }
2135
2136 }
2137
2138 }
2139
2139
2140 }
2141
2142 }
2143
2144 }
2145
2146 }
2147
2148 }
2149
2149
2150 }
2151
2152 }
2153
2154 }
2155
2156 }
2157
2158 }
2159
2159
2160 }
2161
2162 }
2163
2164 }
2165
2166 }
2167
2168 }
2169
2169
2170 }
2171
2172 }
2173
2174 }
2175
2176 }
2177
2178 }
2179
2179
2180 }
2181
2182 }
2183
2184 }
2185
2186 }
2187
2188 }
2189
2189
2190 }
2191
2192 }
2193
2194 }
2195
2196 }
2197
2198 }
2199
2199
2200 }
2201
2202 }
2203
2204 }
2205
2206 }
2207
2208 }
2209
2209
2210 }
2211
2212 }
2213
2214 }
2215
2216 }
2217
2218 }
2219
2219
2220 }
2221
2222 }
2223
2224 }
2225
2226 }
2227
2228 }
2229
2229
2230 }
2231
2232 }
2233
2234 }
2235
2236 }
2237
2238 }
2239
2239
2240 }
2241
2242 }
2243
2244 }
2245
2246 }
2247
2248 }
2249
2249
2250 }
2251
2252 }
2253
2254 }
2255
2256 }
2257
2258 }
2259
2259
2260 }
2261
2262 }
226
```

The screenshot shows the Mbed Studio interface with the following details:

- File Bar:** File Edit Selection View Go Terminal Help
- Left Sidebar:**
  - Active program: mbed-os-example-for-azure
  - Target: DISCO-L475VG-IOT01A (B-L475E-IOT01A)
  - Build profile: Debug
  - Buttons: Run, Stop, Refresh
  - Project Tree:
    - mbed-os-example-blinky
    - mbed-os-example-for-azure
    - .github
    - BSP\_B-L475E-IOT01
    - BUILD
    - drivers
    - mbed-client-for-azure
    - mbed-os
    - ntp-client
    - resources
    - tests
    - mbed
    - azure\_cloud\_credentials.h
    - CMakeLists.txt
    - CONTRIBUTING.md
    - LICENSE
    - main.cpp (highlighted in blue)
    - mbed\_app.json
- Central Area:**
  - File: README.md, main.cpp, azure\_cloud\_credentials.h
  - Code Editor (main.cpp):
 

```
22 // Include all sensor headers
23 #include "stm32l475e_iot01.h"
24 #include "stm32l475e_iot01_tsensor.h"
25 #include "stm32l475e_iot01_hsensor.h"
26 #include "stm32l475e_iot01_psensor.h"
27 #include "stm32l475e_iot01_magnto.h"
28 #include "stm32l475e_iot01_accelero.h"
29 #include "stm32l475e_iot01_gyro.h"
30
31 // Global symbol referenced by the Azure SDK's port for Mbed OS
32 NetworkInterface *_defaultSystemNetwork;
33
34
```
  - Terminal:
 

```
Info: Message sent successfully
Info: Sending sensor data: {"temperature": 24.85, "humidity": 53.42, "pressure(hPa)": 1013.25, "magnetometer_x(mG)": 287, "magnetometer_y(mG)": 167, "magnetometer_z(mG)": -347, "gyroscope_x(dps)": -1820.00, "gyroscope_y(dps)": -70.00, "gyroscope_z(dps)": -560.00, "accelerometer_x(mg)": 269, "accelerometer_y(mg)": 159, "accelerometer_z(mg)": 969}
Info: Message sent successfully
Info: Sending sensor data: {"temperature": 24.96, "humidity": 52.88, "pressure(hPa)": 1012.86, "magnetometer_x(mG)": 283, "magnetometer_y(mG)": 179, "magnetometer_z(mG)": -349, "gyroscope_x(dps)": -700.00, "gyroscope_y(dps)": -350.00, "gyroscope_z(dps)": -70.00, "accelerometer_x(mg)": 272, "accelerometer_y(mg)": 161, "accelerometer_z(mg)": 968}
Info: Message sent successfully
Info: Sending sensor data: {"temperature": 25.03, "humidity": 52.46, "pressure(hPa)": 1012.68, "magnetometer_x(mG)": 286, "magnetometer_y(mG)": 176, "magnetometer_z(mG)": -351, "gyroscope_x(dps)": -560.00, "gyroscope_y(dps)": -350.00, "gyroscope_z(dps)": -210.00, "accelerometer_x(mg)": 267, "accelerometer_y(mg)": 161, "accelerometer_z(mg)": 970}
Info: Message sent successfully
Info: Sending sensor data: {"temperature": 25.14, "humidity": 51.92, "pressure(hPa)": 1012.72, "magnetometer_x(mG)": 289, "magnetometer_y(mG)": 171, "magnetometer_z(mG)": -349, "gyroscope_x(dps)": 210.00, "gyroscope_y(dps)": -980.00, "gyroscope_z(dps)": -1120.00, "accelerometer_x(mg)": 264, "accelerometer_y(mg)": 149, "accelerometer_z(mg)": 974}
Info: Message sent successfully
Info: Sending sensor data: {"temperature": 25.21, "humidity": 51.65, "pressure(hPa)": 1013.03, "magnetometer_x(mG)": 287, "magnetometer_y(mG)": 176, "magnetometer_z(mG)": -350, "gyroscope_x(dps)": -420.00, "gyroscope_y(dps)": -350.00, "gyroscope_z(dps)": -280.00, "accelerometer_x(mg)": 272, "accelerometer_y(mg)": 155, "accelerometer_z(mg)": 968}
Info: Message sent successfully
```
- Bottom Status Bar:** Ln 202, Col 1 LF UTF-8 Spaces:4 C++ 4

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.

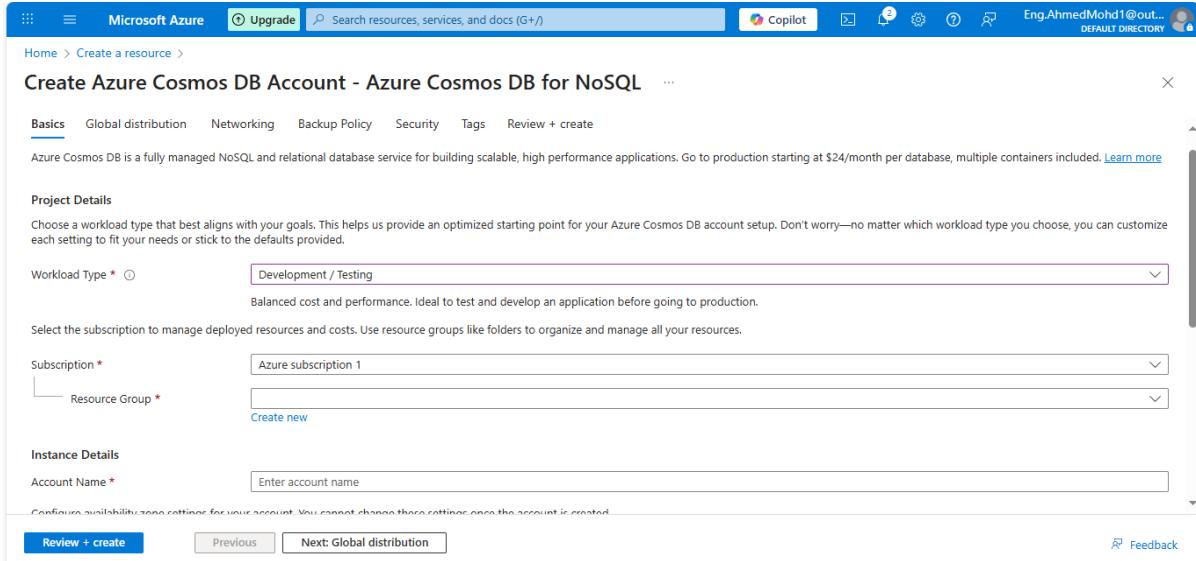


Figure X. Initial setup for creating an Azure Cosmos DB account for NoSQL in the Azure Portal.

- 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

- 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.

- 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”

The screenshot shows the 'Add a route' configuration page in the Azure IoT Hub. The 'Endpoint' tab is selected. The configuration fields are as follows:

- Endpoint type:** Cosmos DB
- Endpoint name:** cosmosEndpoint
- Cosmos DB account:** myiotcosmosdb
- Database:** IoTData
- Container:** Telemetry
- Generate a synthetic partition key for messages:** Disabled (radio button selected)

At the bottom, there is a 'Create + next' button.

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

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

The screenshot shows the 'Add a route' configuration page in the Azure IoT Hub. The 'Route' tab is selected. The configuration fields are as follows:

- Name:** telemetryToCosmos
- Enable route:** Checked
- Data source:** Device Telemetry Message
- Routing query:** 1 true
- Test:** A sample message tests your route query. Results will show whether the sample matched the query or not, and will verify that your query syntax

At the bottom, there are buttons for 'Previous', 'Create + skip enrichments' (which is highlighted in blue), and 'Create + add enrichments'.

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

- 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.

The screenshot shows the Microsoft Azure Data Explorer interface for the 'myiotcosmosdb' database. The left sidebar lists various database management options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Quick start, and Data Explorer. The Data Explorer section is selected. Under Telemetry, there are two main sections: Items and Settings. The Items section displays a list of documents with IDs such as 'id: b92c5304-fe9a-4292-b278-...', 'id: ea21e06c-691a-4104-b87a-...', and 'id: 85b33334-4e4f-4ed1-9e18-...'. A specific document is selected, showing its detailed properties. The properties include 'iothub-consecutive-type': 'upstream', 'iothub-content-encoding': 'utf-8', 'iothub-enqueuedtime': '2025-07-25T20:40:28.028000Z', 'iothub-message-source': 'Telemetry', 'iothub-name': 'arm-uttyler-hcc', and a large JSON object for the 'Body' field containing sensor data like temperature, humidity, pressure, magnetometer values, gyroscope values, and accelerometer values. The JSON body starts with:

```

    "Body": {
        "temperature(C)": 24.9,
        "humidity(%)": 49.03,
        "pressure(hPa)": 1009.59,
        "magnetometer_x(mG)": -280,
        "magnetometer_y(mG)": -199,
        "magnetometer_z(mG)": -691,
        "gyroscope_x(dps)": -350,
        "gyroscope_y(dps)": -490,
        "gyroscope_z(dps)": -280,
        "accelerometer_x(mg)": -14,
        "accelerometer_y(mg)": -25,
        "accelerometer_z(mg)": 1026
    },
    "_id": "ZPdtANw-604E000000000000",
    "_self": " dbs/ZPdtAA=/colls/ZPdtANw-604E000000000000/docs/ZPdtANw-604E000000000000"
}

```

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 ( $180^\circ$ ) for optimal exposure.
- Angle the panel to match local latitude ( $\approx 30^\circ$  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<sub>4</sub> 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 ( $180^\circ$ ), ensuring proper orientation for maximum solar efficiency in the northern hemisphere.



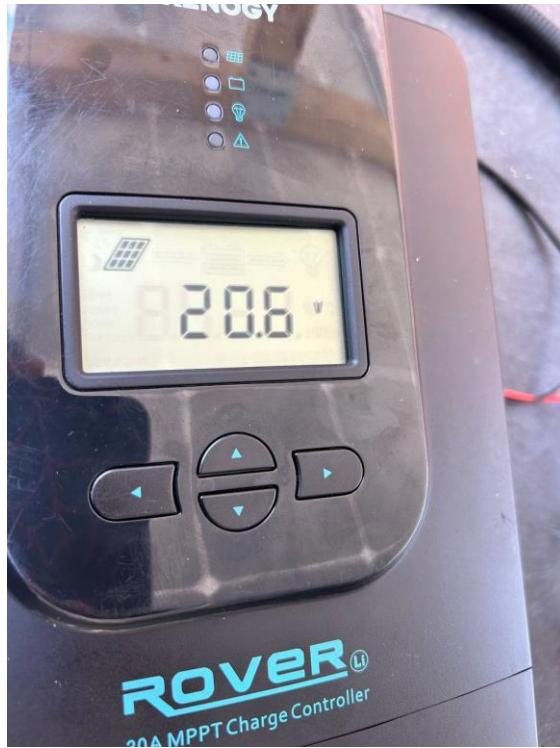
Figure. A 15A DC circuit breaker (DIHOOL DZ47X-63) is installed between the solar panel and charge controller for overcurrent protection and system safety, it is switched on.

## 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.

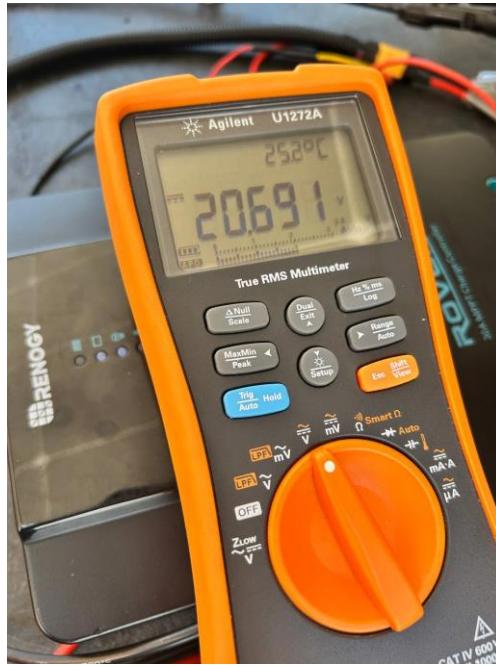


Figure Measurement of open-circuit voltage (Voc) of the solar panel under full sunlight using an Agilent U1272A multimeter. The reading shows 20.691 V, confirming proper panel output.

## 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.

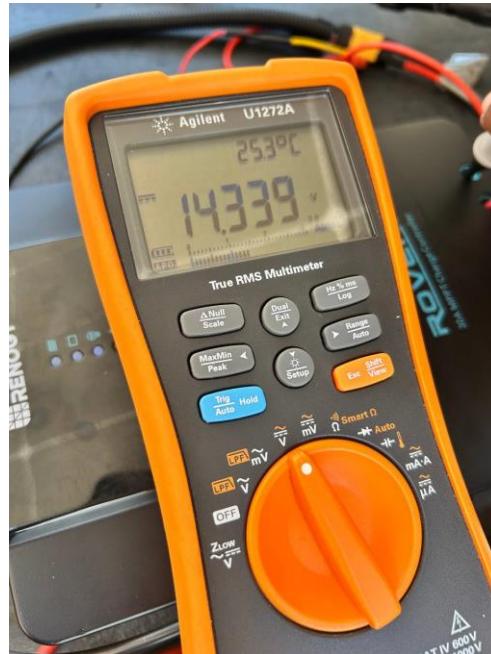


Figure. An Agilent U1272A multimeter confirms the charge controller's output by measuring a voltage of 14.339 V at the battery connection, validating that the controller is delivering appropriate charging voltage.

## 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.

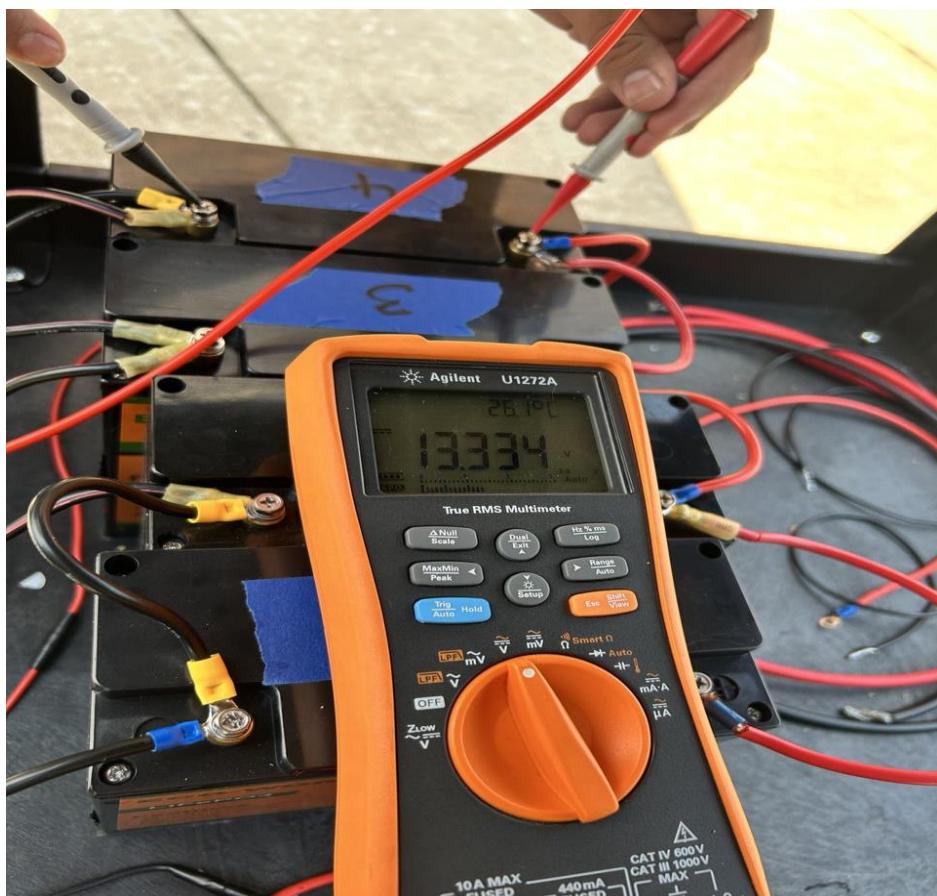


Figure. The multimeter shows 13.334 V, indicating the voltage under load conditions—slightly elevated due to active charging or recent current draw.

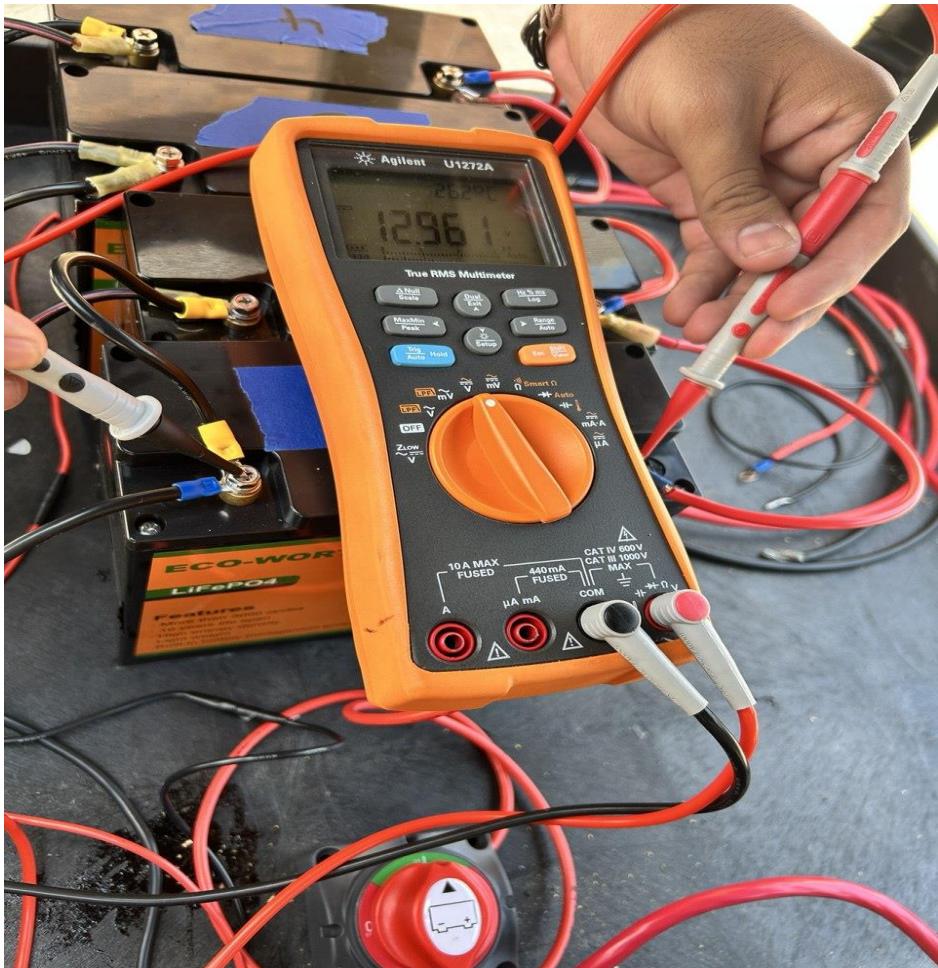


Figure. The multimeter reads 12.961 V, which represents the resting voltage of the battery bank under no active load.

### 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.

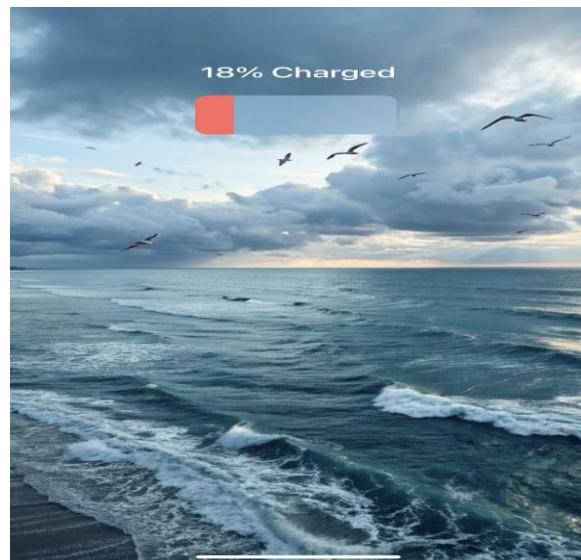


Figure. A connected device shows 18% charged, confirming successful AC output from the inverter to the load via USB or outlet connection.



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 = readINAResister24(ina_addrs[i], REG_VBUS);
    float voltage = rawV * 0.1953125 / 1e6 * VBUS_SCALE;

    int32_t rawI = readINAResister24(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("-----");
}

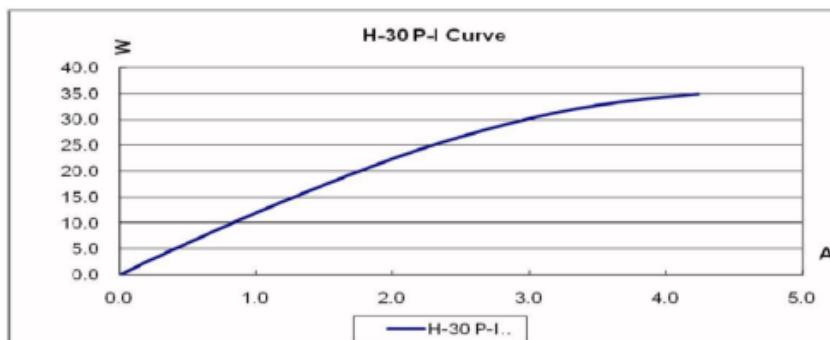
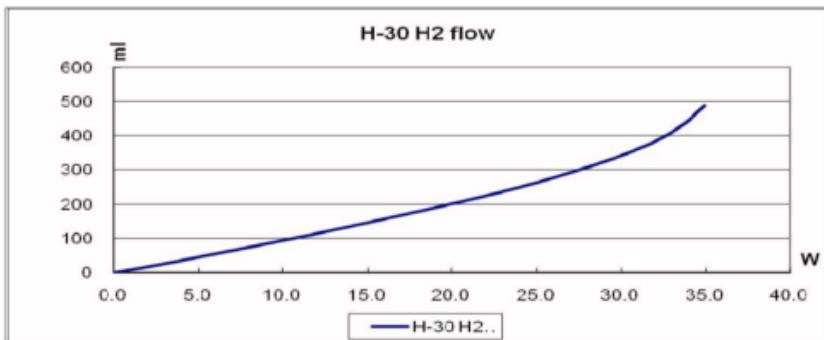
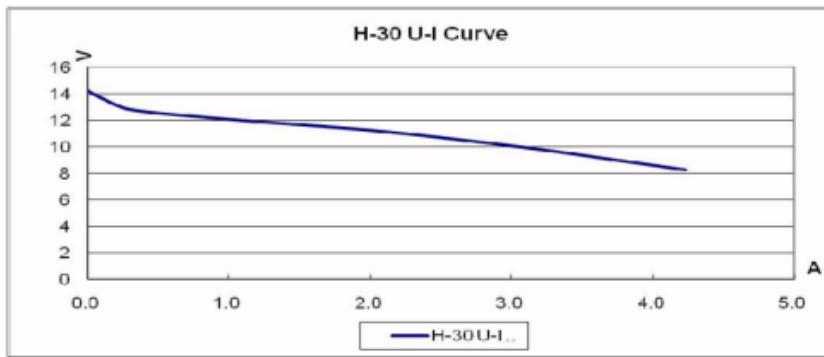
```

## 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 30°C
Max stack temperature	55°C
H2 Pressure	0.45-0.55bar
Hydrogen purity	≥ 99.995 % dry H2
Humidification	self-humidified
<u>Cooling</u>	<u>Air (integrated cooling fan)</u>
Weight (with fan & casing)	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:**