Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

Team 4: Automatic Solar Lighting System Bi-Weekly Update 4

Atahan Bakanyildiz, Romi Gilat, Cedar Maxwell, Nick Miller

Sponsor: Wohnyeok Jang
TA: Fahrettin Ay

What is the Automatic Solar Lighting System?

Problem:

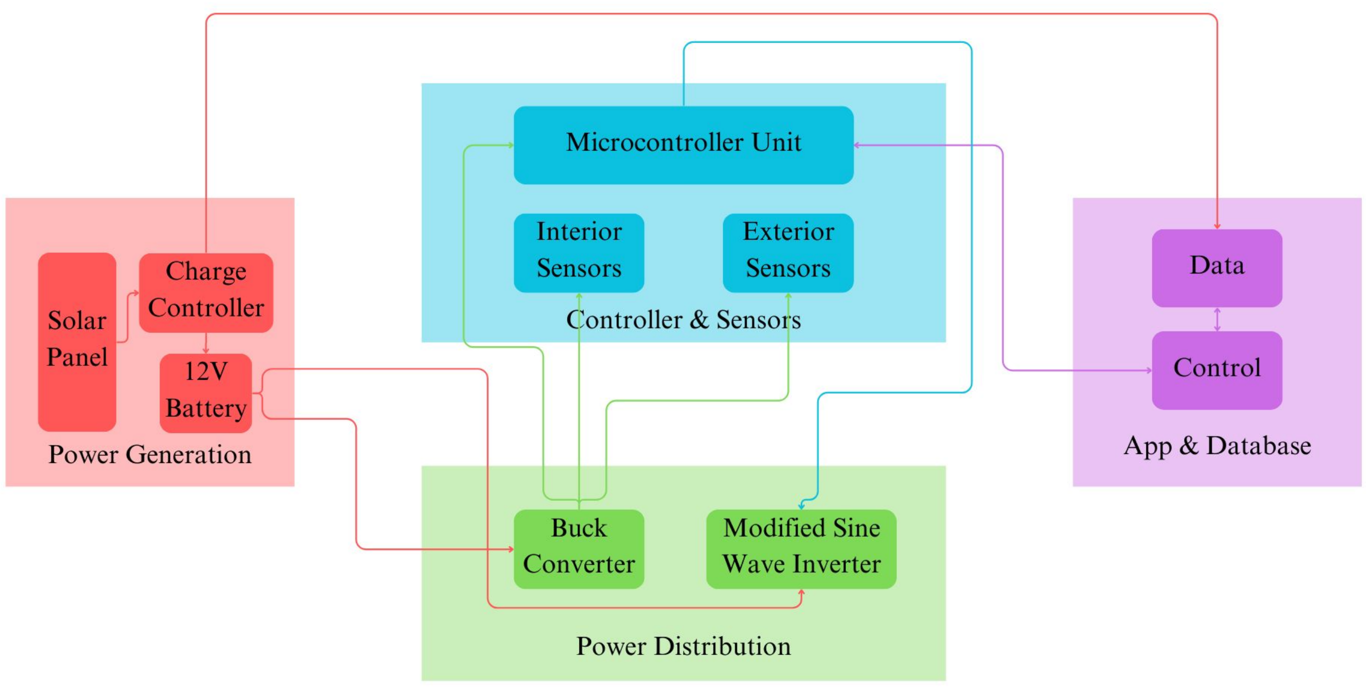
- Increasing grid demand and reliability issues make power outages more frequent.
- Integrating solar energy into homes ensures a dependable and independent power source.
- Solar power provides backup energy during outages, reduces electricity costs, and supports sustainability.

Solution:

- Solar panels provide up to 3 days of reliable power during emergencies.
- Motion-activated lighting for the foyer and patio ensures security and efficiency.
- Remote control via an app allows convenient access and management of the system.
- Enhances energy independence, cost savings, and sustainability for homeowners.

System Visual

Project/Subsystem Overview



Nick: Power Distribution

Romi: Controller & Sensors

Atahan: Power Generation

Cedar: App & Database

Project Timeline

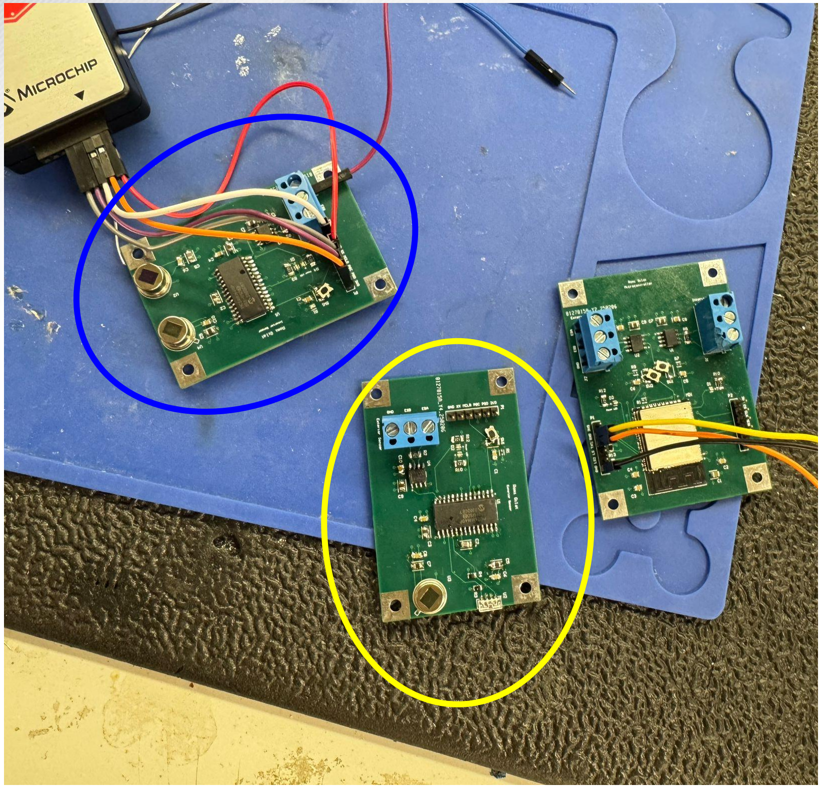
Subsystems designed and completed testing (Complete)	Integration of MCU and Power generation (to complete by 3/7)	Integration of MCU and Android App (to complete 3/7)	Integration of MCU and power distribution (to complete by 3/7)	System Test (to complete by 3/21)	Validation (to complete by 3/21)	Demo and Report (to complete by 4/28)

Controller & Sensors

Romi Gilat

Accomplishments since last update 16 hrs of effort	Ongoing progress/problems and plans until the next presentation
• Attempted integration with all team members • Tested code with Atahan for power monitor • Finished PCB for interior and exterior sensors	• Attend Blitz • Finalize sensor code upload sequence • Finalize communication between MCU & sensors • Continue integration

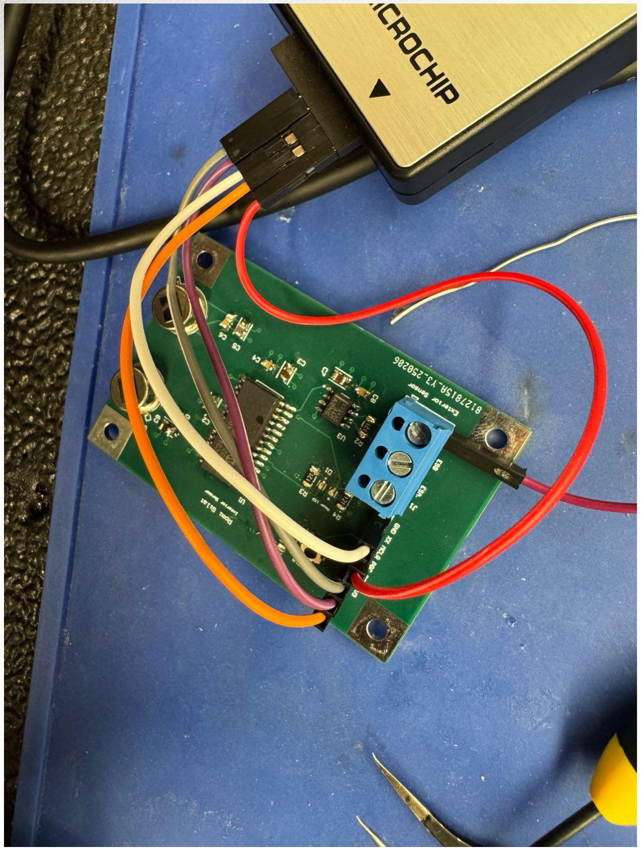
Sensors

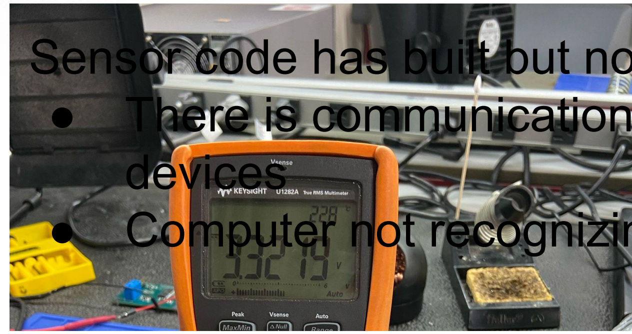


Interior & exterior sensors are complete

- Running at constant 3.3 V
- Consistent ~ 0.05 - 0.02A

Coding Sensors



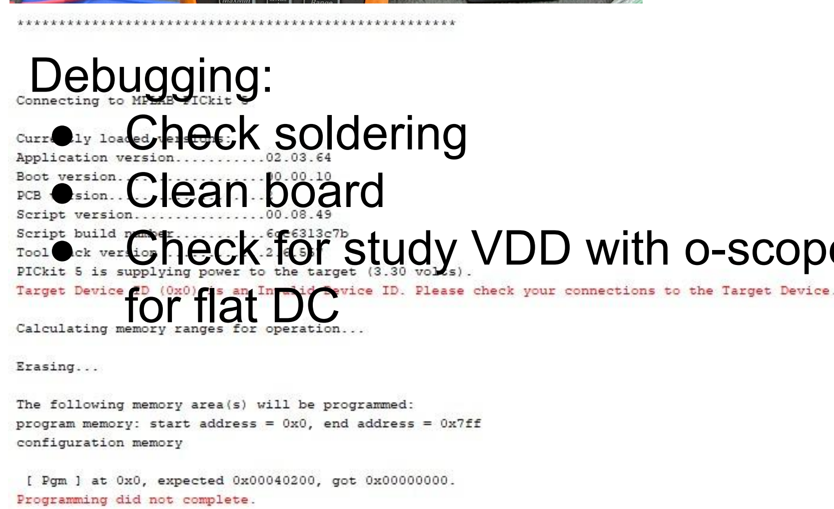


Sensor code has built but not integrated

- There is communication between devices
- Computer not recognizing

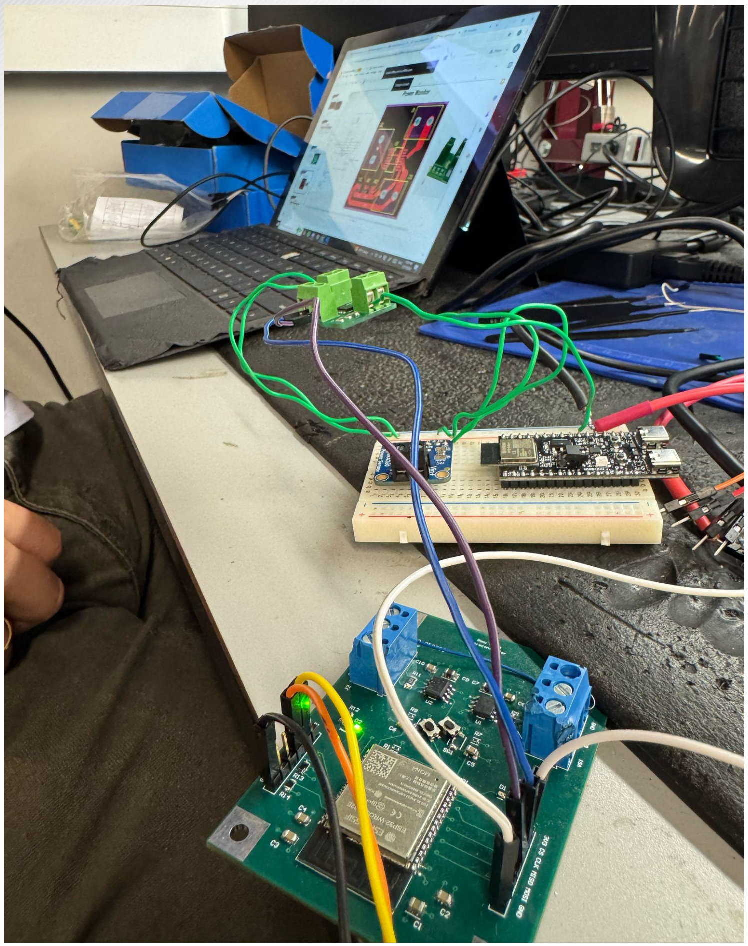
Debugging:

- Check soldering
- Clean board
- Check for study VDD with o-scope for flat DC



```
*****************************************************

Connecting to [illegible]

Curr[illegible]ly loa[illegible]:
Application version...........02.03.64
Boot version..............[illegible]0.00.10
PCB [illegible]sion..[illegible]
Script version................00.08.49
Script build [illegible]..........6[illegible]6313c7b
Tool[illegible]ck ver[illegible]
PICkit 5 is supplying power to the target (3.30 vo[illegible]s).
Target Device [illegible] (0x0) [illegible]s an In[illegible] Device ID. Please check your connections to the Target Device.

Calculating memory ranges for operation...

Erasing...

The following memory area(s) will be programmed:
program memory: start address = 0x0, end address = 0x7ff
configuration memory

 [ Pgm ] at 0x0, expected 0x00040200, got 0x00000000.
Programming did not complete.
```

Integration with Power Generation



What worked:

- Integrating new code into MCU
- Added multiple test cases to check for registers

What didn’t:

- Tested many versions of code, landed on testing for register readability which failed
- Not recognizing registers with both OTS and designed power monitor

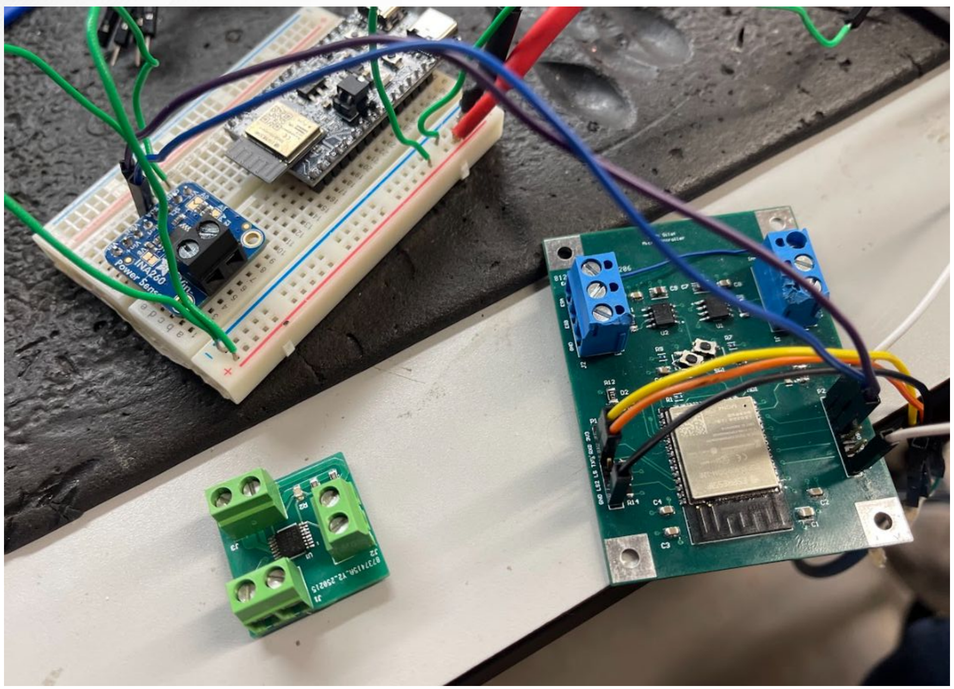
Power Generation

Atahan Bakanyildiz

Accomplishments since last update 16 hrs of effort	Ongoing progress/problems and plans until the next presentation
• System integrated with power distribution • Power monitor partially integrated with MCU	• Power Monitor needs full integration with MCU • Power Monitor data on MCU needs to be integrated with app

Power Generation

Power Monitor -> MCU



Integration process with MCU for Power Monitor:

- Tested with off-the-shelf and with the PCB version

Pictured is testing with the off the shelf version of the power monitor.

Power Generation

Power Monitor -> MCU

```
C rs485_example.c M ×    ESP-IDF Welcome    C ina260[1].c
main > C rs485_example.c > ...
void read_ina260_data() {
    if (read_register(INA260_REG_POWER, &power_raw) == ESP_OK) {
    }
}

// Main function
void app_main() {
    i2c_master_init();
    vTaskDelay(pdMS_TO_TICKS(1000)); // Delay to allow I2C device initialization

    while (1) {
        read_ina260_data(); // Read and print INA260 sensor values
        vTaskDelay(pdMS_TO_TICKS(2000)); // Delay between reads
    }
}
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

I (240) cpu_start: cpu freq: 160000000 Hz
I (240) app_init: Application information:
I (243) app_init: Project name:     echo_rs485
I (248) app_init: App version:      0dbc2a8-dirty
I (254) app_init: Compile time:     Mar 17 2025 18:01:35
I (260) app_init: ELF file SHA256:  9dfaab0b2...
I (265) app_init: ESP-IDF:          v5.3.1
I (270) efuse_init: Min chip rev:     v0.0
I (275) efuse_init: Max chip rev:     v3.99
I (280) efuse_init: Chip rev:         v3.1
I (285) heap_init: Initializing. RAM available for dynamic allocation:
I (292) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (298) heap_init: At 3FFB2C48 len 0002D3B8 (180 KiB): DRAM
I (304) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (310) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (317) heap_init: At 4008CF98 len 00013068 (76 KiB): IRAM
I (324) spi_flash: detected chip: generic
I (328) spi_flash: flash io: dio
I (332) main_task: Started on CPU0
I (342) main_task: Calling app_main()
I (342) INA260_I2C: Initializing I2C...
I (342) gpio: GPIO[21]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (352) gpio: GPIO[22]| InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
E (362) i2c.master: i2c_master_bus_add_device(990): invalid scl frequency
E (362) INA260_I2C: Failed to add INA260 device: ESP_ERR_INVALID_ARG
E (1372) INA260_I2C: I2C device handle is NULL. Cannot read.
E (1372) INA260_I2C: I2C device handle is NULL. Cannot read.
```

Initially the MCU could not find the I2C slave device of the power monitor.

- Address was 100000
- Issue was resolved with master frequency:
  - Initially code was set at 100000 Hz
  - Address connected with 115200 Hz as set on MCU

```
// I2C configuration
#define I2C_MASTER_NUM              I2C_NUM_0   // I2C port number for master dev
#define I2C_MASTER_SCL_IO           22          // GPIO number for I2C master clock
#define I2C_MASTER_SDA_IO           21          // GPIO number for I2C master data
#define I2C_MASTER_FREQ_HZ          115200      // I2C master clock frequency
#define I2C_MASTER_TX_BUF_DISABLE   0           // I2C master doesn't need buffer
#define I2C_MASTER_RX_BUF_DISABLE   0           // I2C master doesn't need buffer
```

Power Generation

Power Monitor -> MCU

Register on slave for current

```
// INA260 I2C address and register definitions
#define INA260_ADDR              0x40       // Default INA260 address
#define INA260_REG_CONFIG        0x00
#define INA260_REG_CURRENT       0x01
#define INA260_REG_BUSVOLTAGE    0x02
#define INA260_REG_POWER         0x03
#define INA260_REG_MASK_ENABLE   0x06
```

Register on slave for voltage

Register on slave for power

These registers are not being found by the MCU.
The code makes calculations and stores V, A, and W for app communication

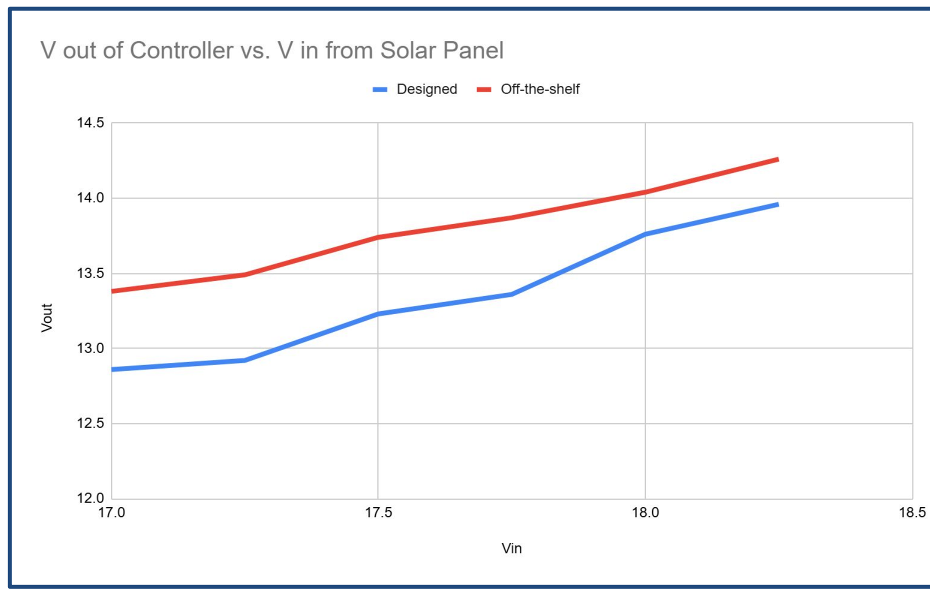
```
while (1) {
    // Read current register
    if (ina260_read_reg(INA260_REG_CURRENT, &current_raw) == ESP_OK) {
        ESP_LOGI(TAG, "Current Raw: 0x%04x", current_raw);
    }
    // Read bus voltage register
    if (ina260_read_reg(INA260_REG_BUSVOLTAGE, &voltage_raw) == ESP_OK) {
        ESP_LOGI(TAG, "Bus Voltage Raw: 0x%04x", voltage_raw);
    }
    // Read power register
    if (ina260_read_reg(INA260_REG_POWER, &power_raw) == ESP_OK) {
        ESP_LOGI(TAG, "Power Raw: 0x%04x", power_raw);
    }
    // Delay for a second before reading again
    vTaskDelay(pdMS_TO_TICKS(1000));
```

Power Generation

Charge Controller -> Power Distribution

Charge Controller:

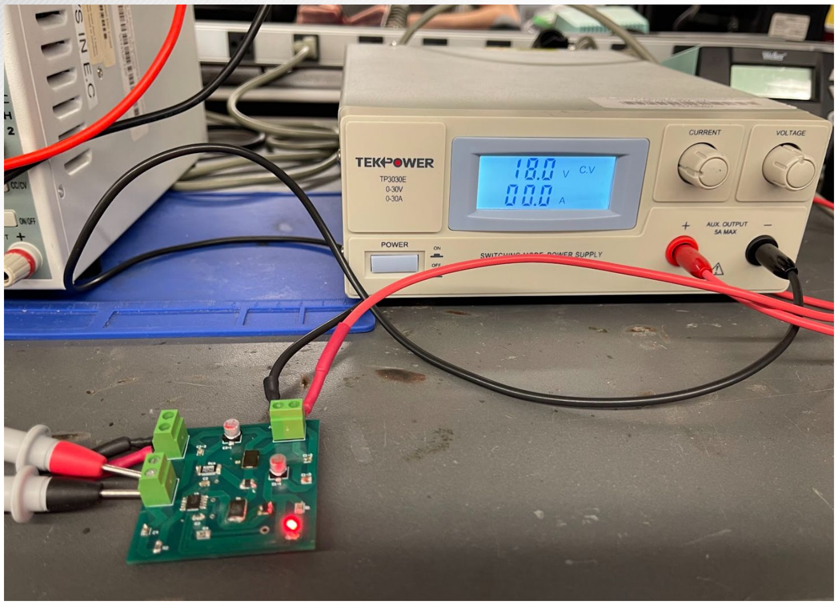
- Tested with power supply (emulate solar panel V)
- Same tests with off-the-shelf component.
- 3.56% discrepancy between the two.
- Electronic Load of 13.7 V (Battery peak voltage).



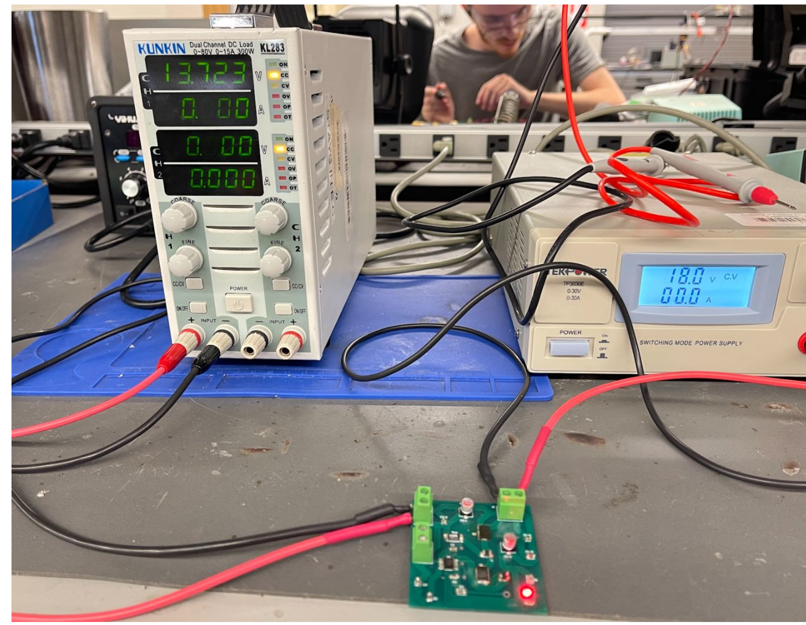
Integration:

- Works fully with 12V battery.
- E-load represents draw from power distribution
- Same results as previous

Charge Controller Integration



No load with 18V "Solar Panel"
Red LED means fully charged (Overcharge Protection is on)



13.7 V load with 18V "Solar Panel"
Red LED is still on meaning overcharge protection occurring even with load on system

App & Database

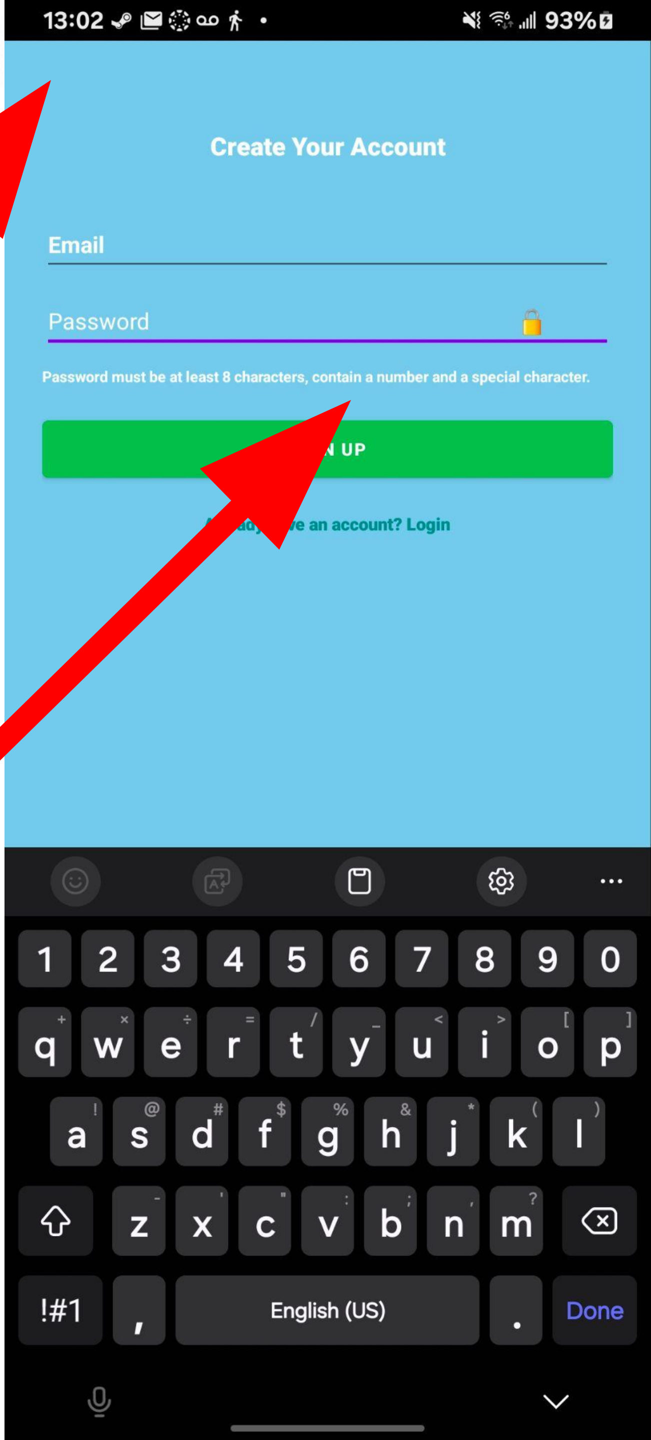
Cedar Maxwell

Accomplishments since last update 20 hrs of effort	Ongoing progress/problems and plans until the next presentation
• Added stricter password requirements to the app • Improved app Home Screen design to show more useful information instead of just “log out” • Graph can show daily, weekly and monthly data now, instead of just data since opening the app • Added user-customizable names for lights	• App remembering user on force close and scheduling of light toggling still in progress • Firebase ESP32 code integration with MCU ESP32 code unfinished

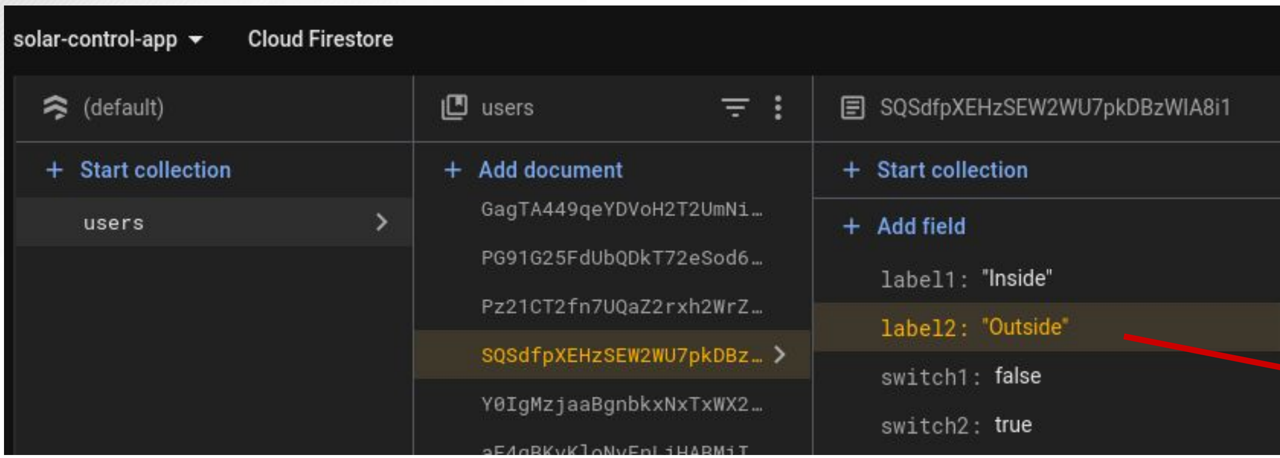
Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

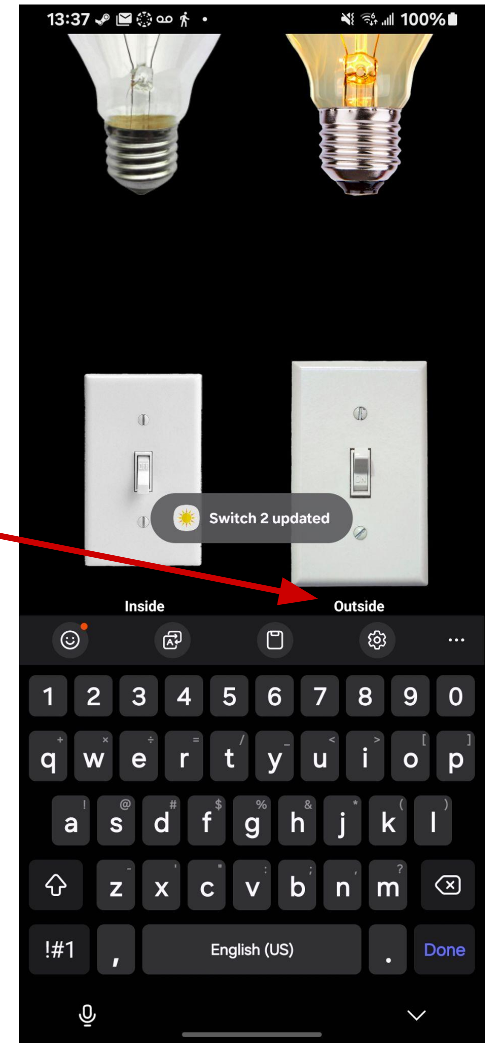
App Login Improvements

- Removed Title bar atop app
- Stricter password requirements now enforced
- Helpful message displayed beneath password entry box
- Added “Lock” button to hide/show password characters



Customizable Switch Labels





- Switch labels are now user-customizable
- Customized labels are saved per-user, synced with Firebase and persist between logins

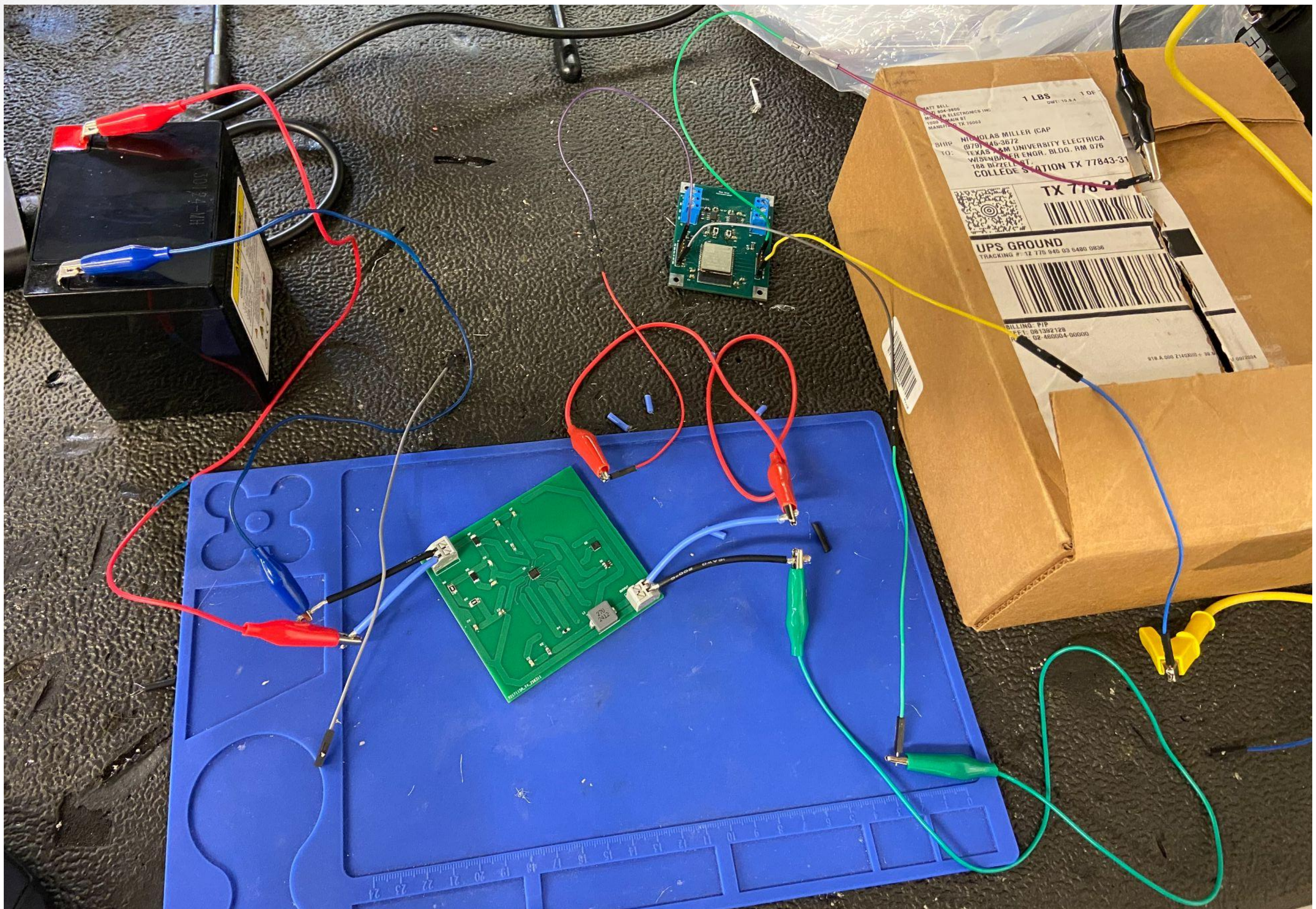
Dwight Look College of ENGINEERING TEXAS A&M UNIVERSITY

Power Distribution

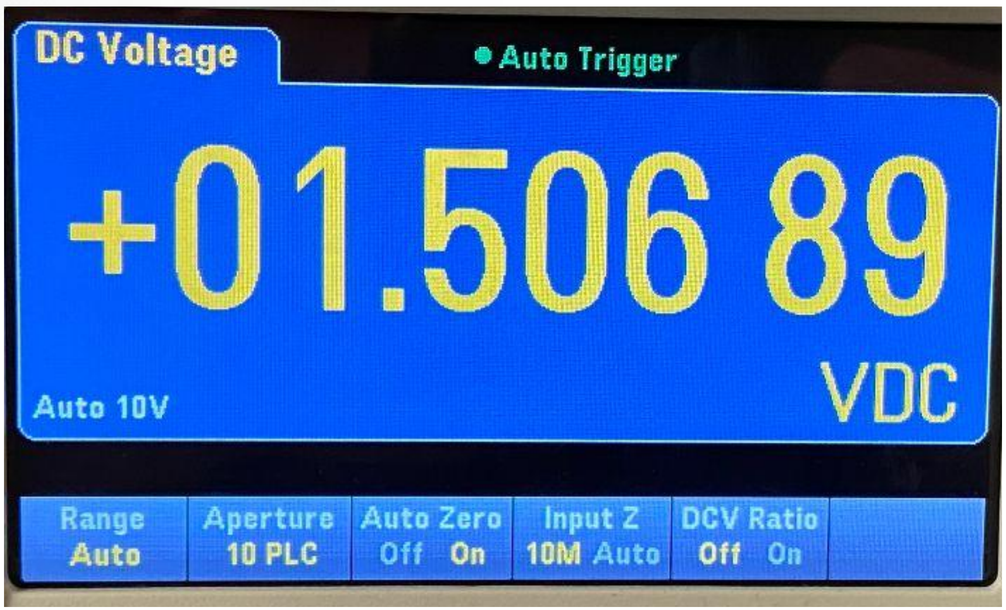
Nick Miller

Accomplishments since last update 26 hrs of effort	Ongoing progress/problems and plans until the next presentation
• Successfully integrated with Power Distribution. • Attempted integration with MCU & Sensors to achieve partial integration.	• Significant voltage drop with Buck Converter when connected to a load. • Attend Blitz. • Finalize integration with MCU & Sensors.

Integration with Power Generation



Integration with MCU & Sensors



What worked:

- Able to connect from Buck Converter output to MCU input.
- Able to provide a steady output for MCU.

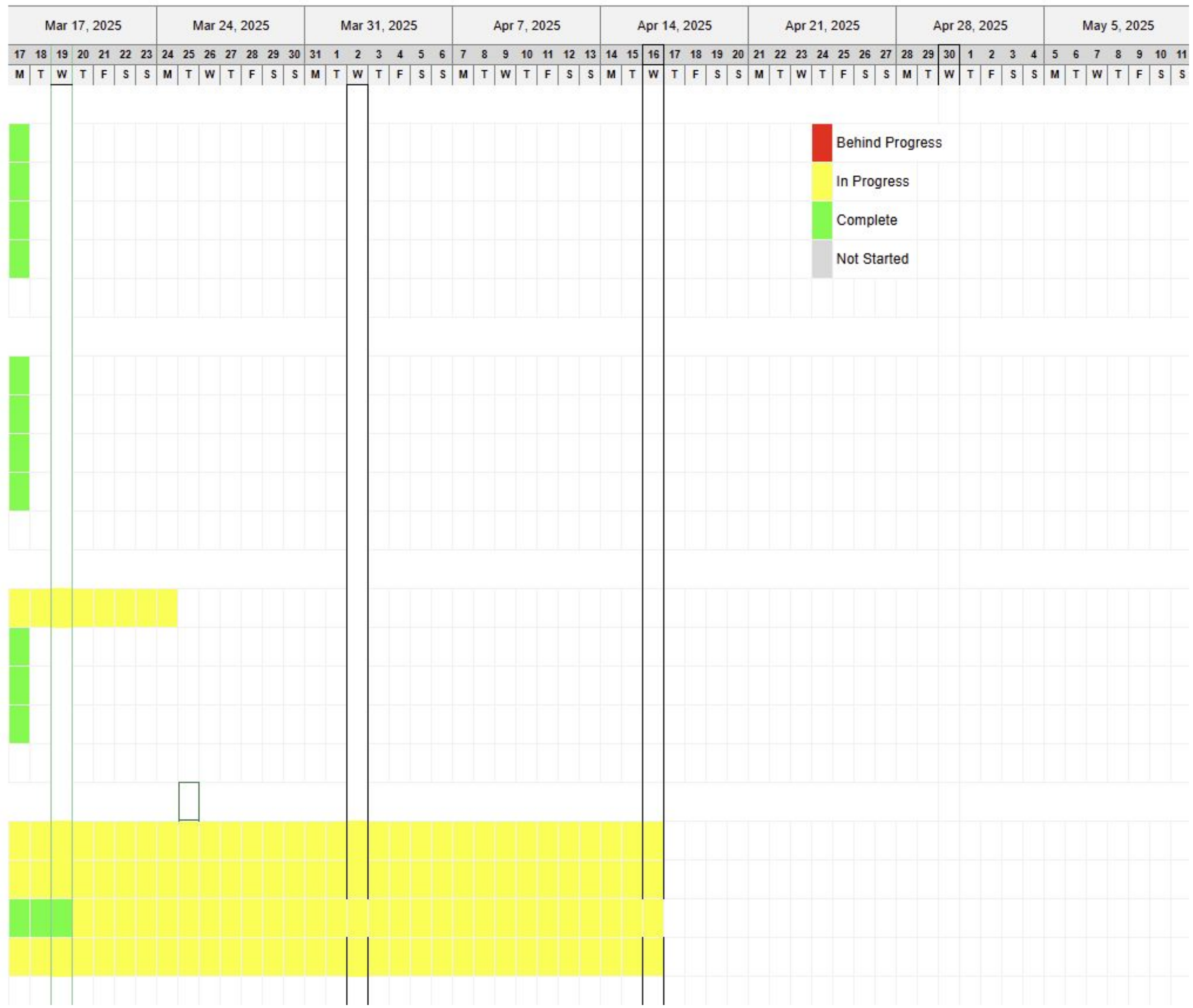
What didn't:

- When connected to MCU, output voltage significantly drops to 1.506V.

Power Distribution

Execution & Validation Plan

TASK	ASSIGNED TO	PROGRESS	START	END
Testing Subsystems				
Tested Subsystem	Romi Gilat	100%	1/15/25	1/22/25
Test Subsystem	Nick Miller	100%	1/29/25	2/12/25
Tested Subsystem	Atahan Bakanyildiz	100%	1/15/25	1/19/25
Tested App and Database	Cedar Maxwell	100%	1/13/25	1/13/25
Completion Progress	Team	100%	1/15/25	1/25/25
Subsystem Redesign				
PCB Redesign	Romi Gilat	100%	1/23/25	2/5/25
Circuit and PCB Redesign	Nick Miller	100%	1/13/25	1/13/25
PCB Design and Test	Atahan Bakanyildiz	100%	1/13/25	1/16/25
N/A	Cedar Maxwell	100%	1/13/25	1/13/25
Completion Progress	Team	100%	1/13/25	1/16/25
Subsystem validation				
Build, Test & Validate	Romi Gilat	80%	2/5/25	3/24/25
Build, Test & Validate	Nick Miller	100%	2/17/25	3/4/25
Build PCB and Validate	Atahan Bakanyildiz	100%	2/22/25	2/27/25
App Completed and Synced with Firebase	Cedar Maxwell	100%	1/13/25	1/13/25
Completion Progress	Team	90%	2/22/25	3/24/25
Subsystem Integration				
Integrate subsystems	Romi Gilat	25%	3/24/25	4/13/25
Integrate with MCU/Sensors	Nick Miller	60%	2/20/25	3/7/25
Integrate with MCU and Distribution	Atahan Bakanyildiz	50%	3/8/25	3/11/25
Firebase Connected with MCU	Cedar Maxwell	75%	1/13/25	3/7/25
Completion Progress	Team	50%	3/31/25	4/25/25



Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

Validation Plan

Standby Wake-Up Miss Rate	The maximum number of miss trigger incidents within the sensor's field of view will be 15% or less.	UNTESTED	Romi Gilat
False Positive Rate	Within the sensor system, the false positive rate will be less than 15% in case of small animals or critters walking within the range.	UNTESTED	Romi Gilat
Battery Operating Time	The operating time of the 12V Lead-Acid battery shall be between 10 and 20 hours.	UNTESTED	Nick Miller
Solar Charging Time	The solar charging time shall be between 4 and 6 hours.	SUCCESS	Atahan
System Area	The system area shall include the rooftop, foyer and exterior of a household.	SUCCESS	Atahan
Installation	The solar panel installation will be done up to National Electrical Code (NEC), International Building Code (IBC), and International Fire Code (IFC), along with the mounting of the system 45 degrees tilted offset from the ground level for optimal sunlight units.	UNTESTED	All
Mounting	The automatic solar lighting system includes **solar panels**, an **indoor sensor with integrated lights**, and an **outdoor sensor with lights**. The lightweight design eliminates the need for structural support, and the solar panels are roof-mounted for optimal sunlight exposure	UNTESTED	All
Inputs	The Automatic Solar Lighting System processes multiple inputs across subsystems. The **Power Generation** subsystem captures solar energy via photovoltaic panels, while the **Buck Converter/Inverter** adjusts DC voltage for the MCU and lighting. The app monitors light status and subsystem performance in real time.	Partial Success	Atahan, Nick, Romi

Validation Plan continued

Power Consumption	The system shall consume approximately 18 Watts, 9 Watts per light bulbs.	UNTESTED	Nick Miller
Input Voltage Level	The input voltage level shall be +10 VDC to +14 VDC.	SUCCESS	Nick Miller
External Commands	The Automatic Solar Lighting System shall document all external commands in the appropriate ICD.	UNTESTED	Cedar Maxwell
Data Output	The Automatic Solar Lighting System will output the status of the porch and foyer lights by means of the mobile application.	UNTESTED	Cedar Maxwell
Diagnostic Output	The MCU will transmit diagnostic data to the app for display.	UNTESTED	Cedar Maxwell, Romi Gilat
Connectors	The Automatic Solar Lighting System will use the American National Standard for Electrical Connectors ANSI C119.6-2011.	SUCCESS	All
Wiring	The Automatic Solar Lighting System will follow the guidelines set forth by the National Electrical Code regarding electrical wiring. The standard applications of electrical systems is in the article NFPA 70 (NEC).	UNTESTED	All
Altitude	The Automatic Solar Lighting System shall be able to operate efficiently at altitudes around 300 feet.	UNTESTED	Nick, Romi, Atahan
Thermal	The Automatic Solar Lighting System shall be able to operate efficiently at temperatures ranging from 0℃ to 70℃. The microcontroller unit will be located indoors, where the temperature is expected to range from 0°C to 70°C. The sensor system will be used both indoors and outdoors, and is rated for temperatures from -40°C to 85°C.	UNTESTED	Nick, Romi, Atahan
Humidity	The sensor unit will function up to 90% humidity for proper functioning.The sensors themselves need to be placed in a water proof, sealed container that will prevent the electronics from getting drenched.	UNTESTED	Nick, Romi, Atahan

Thank you!