

AUTOMATED SELF-CLEANING SOLAR PANEL

By

Yousaf Abdul Salam

Terry Green

Yann Tyng Lin

Final Report for ECE 445, Senior Design, Spring 2014

TA: Mustafa Mukadam

Project No. 15

May 7, 2014

Abstract

We designed and built an automated self-cleaning solar panel. The panel detects the presence of an obstruction shading a cell, and actuates a cleaning mechanism that cleans off the obstruction and, therefore, restores the panel to normal capacity. To power the cleaning mechanism, we built our own power supplies which are supplied by a 12V battery. When required, this battery is charged by solar power when the cleaning mechanism is idle.

Contents

1. Introduction	1
1.1 Objectives.....	1
1.1.1 Project Goals:	1
1.1.2 Functions:.....	1
1.1.3 Features:	1
1.2 Overall System Description	2
1.2.1 Microcontroller	2
1.2.2 Power Supply	3
1.2.3 Photovoltaic Panel	3
1.2.4 Motor and Spray Valve Control	3
1.2.5 Differential Measurement Unit (DMU).....	3
1.2.6 Cleaning Mechanism	3
2 Design.....	4
2.1 Microcontroller	4
2.2 PV Panel	4
2.3 Differential Measurement Unit.....	4
2.4 Power Supply	5
2.4.1 Battery Charger	5
2.4.2 Battery.....	6
2.4.3 Logic Power Supply	6
2.4.4 Motor Power Supply	7
2.5 Motor and Valve Control	8
2.5.1 Motor Control	8
2.5.2 Valve Control.....	10
2.6 Cleaning Mechanism.....	10
2.6.1 Wiper.....	10
2.6.2 Sprayer	10
3. Design Verification	11
3.1 Microcontroller	11

3.2 PV Panel	11
3.3 Differential Measurement Unit.....	11
3.4 Power Supply	12
3.4.1 Battery Charger	12
3.4.2 Logic Power Supply	14
3.4.3 Motor Power Supply	14
3.5 Motor and Valve Control	15
3.5.1 Motor Control	15
3.5.2 Valve Control.....	15
3.6 Cleaning Mechanism	16
3.6.1 Wiper.....	16
3.6.2 Sprayer	16
4. Costs.....	17
4.1 Parts	17
4.2 Labor	18
5. Conclusion.....	19
5.1 Accomplishments.....	19
5.2 Uncertainties.....	20
5.3 Ethical considerations	20
5.4 Future work.....	20
References	22
Appendix A Requirement and Verification Table.....	23
Appendix B MSP430 Code.....	29

1. Introduction

There is an urgency in improving the efficiency of solar power generation. Current solar panels setups take a major power loss when unwanted obstructions cover the surface of the panels. The obstruction turns the shaded cell into a resistor, causing it to heat up and consume extra power [1]. To address this issue, we have successfully engineered a self-cleaning solar panel. This specific panel detects the obstruction with a Differential Measurement Unit (DMU). It makes the decision from the Microcontroller unit to either clean the panel with the Wiper and Sprayer Mechanism or continue to charge the battery with the Battery Charger. Our mechanism to combat the power loss is unique, self-reliant, and easy to use.

1.1 Objectives

1.1.1 Project Goals:

- Design a mechanism to detect obstructions on solar panels causing significant loss of power
- Design a cleaning mechanism that runs across the length of the panels
- Improve overall solar panel efficiency

1.1.2 Functions:

- Enables the cleaning mechanism once an obstruction has been detected
- Is able to distinguish between whether the obstruction is partially shading the panel or not (e.g. complete cloud shading)
- Is able to charge the battery when the battery is not full and solar panel is receiving enough sunlight.
- Is Idle when it's not charging or cleaning.

1.1.3 Features:

- One-time installation
- Effective wiping system
- Easily replaceable wipers
- Waterproof
- Adjustable degrees of elevation of panels
- Charging system compatible with regular 12V lead acid batteries.

1.2 Overall System Description

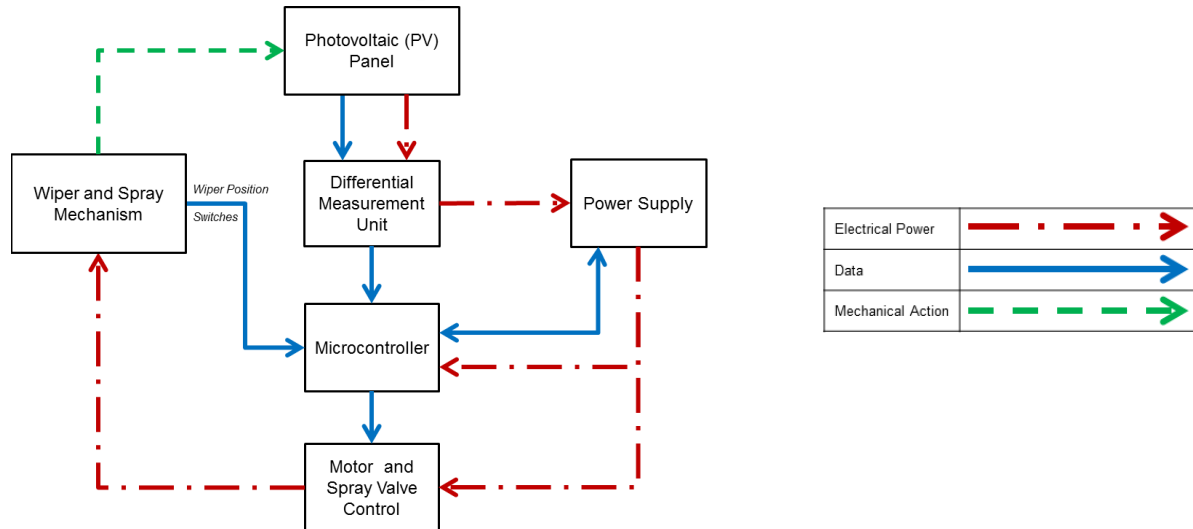


Figure 1. Automated Self-Cleaning Solar Panel Block Diagram (with legend)

1.2.1 Microcontroller

The Microcontroller unit runs according to the algorithm of Figure 2.

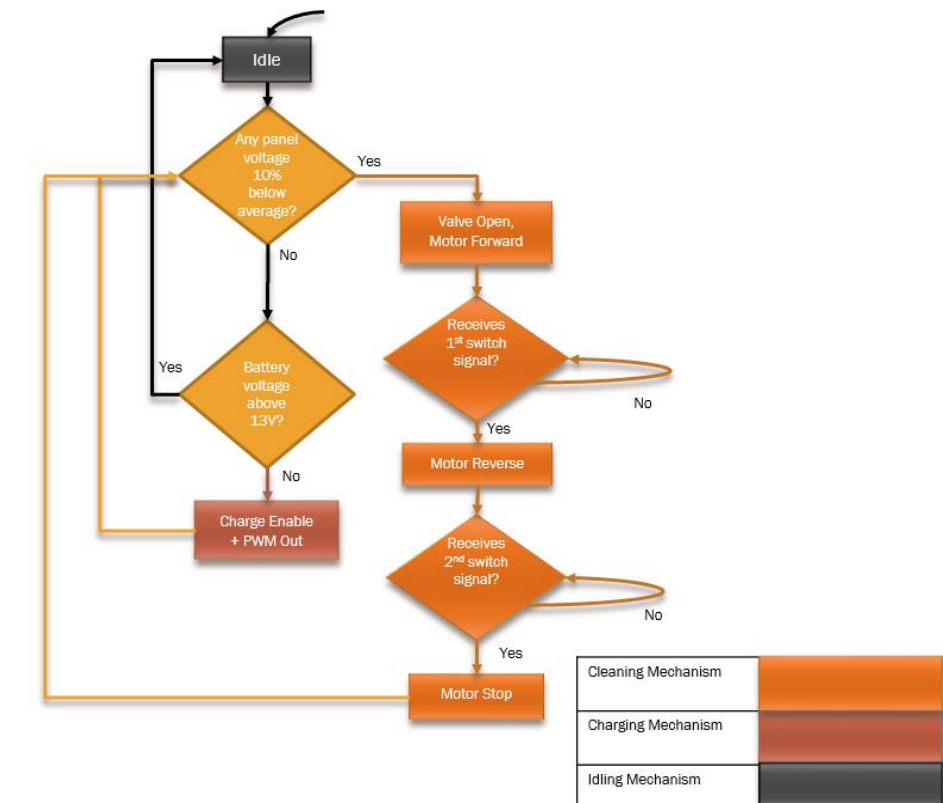


Figure 2. Automated Self-Cleaning Solar Panel Algorithm

1.2.2 Power Supply

The Power Supply unit consist of a battery charger and a 12V lead acid battery. It converts and supplies power to the entire circuit.

1.2.3 Photovoltaic Panel

The PV Panel consists of six solar cells connected in series to provide a $30V_{\text{peakpower}}$ and $0.5A_{\text{peakpower}}$ to charge the Battery. The cells also provide signals to the DMU.

1.2.4 Motor and Spray Valve Control

The Motor and Spray Valve control consist of the solenoid valve and an H-bridge.

1.2.5 Differential Measurement Unit (DMU)

The DMU takes in 6 differential PV cell voltages to convert them to 6 ground referenced PV cell voltages for measurement by the microcontroller. Additionally, the PV panel is the power source for the circuit. The DMU also passes the full panel voltage to the Battery Charger.

1.2.6 Cleaning Mechanism

The Cleaning Mechanism consists of 6 sprayers, a wiper, and a nominal 7.2V DC motor. These are the physical components that clean the panel.

2 Design

2.1 Microcontroller

The Microcontroller chosen is the MSP430G2553. It is specifically chosen because it has 16MHz speed and we need up to 20 I/O pins. It is powered from a 3.3V source.

2.2 PV Panel

The PV cells were chosen to provide the series combination of voltage and current at peak power. Additionally, the PV Panel provides 6 differential voltages from the PV Cells. The differential voltages provide both the signals and source of power to the DMU. A glass and metal frame was manufactured by ECE Machine Shop to protect the PV Cells from the cleaning operation. The spray and wiper mechanism perform the cleaning operation on the PV Panel.

2.3 Differential Measurement Unit

The only method to convert the cell voltages was to use op-amps in either an instrumentation or difference configuration. The difference configuration was chosen since low input resistance was not a concern. Also, the difference configuration used far less resistors and op-amps [2].

The DMU uses 6 operational amplifiers connected in a difference amplifier configuration as shown in Figure 3 [3] [2]. According to Equation 1, R_1 was chosen to be $1.5M\Omega$ to have a large input resistance. If the assumption of Equation 2 is true, each difference amplifier changes the differential voltage input from a PV cell to a ground referenced voltage according to Equation 3. R_2 was chosen based on Equation 4. The 0.5 ratio was selected to keep a maximum 6V PV cell voltage from exceeding the maximum 3.3V input rating of the microcontroller. Figure 4 shows the complete design of the DMU.

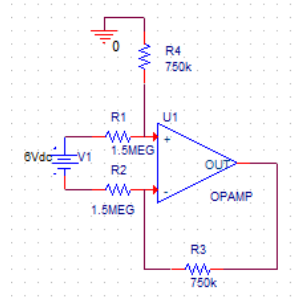


Figure 3. Operational Amplifier in Difference Configuration

$$R_{input} = 2 \times R_1, \quad \text{Make } R_1 \text{ large} \quad (1)$$

$$R_1 = R_2 \text{ and } R_3 = R_4 \quad (2)$$

$$V_{out} = \frac{R_3}{R_1} V_{differential}, \quad \text{Output voltage} \quad (3)$$

$$R_2 = \frac{V_{out,max}}{V_{differential,max}} R_1 = \frac{3V}{6V} \times 1.5M\Omega = 750k\Omega \quad (4)$$

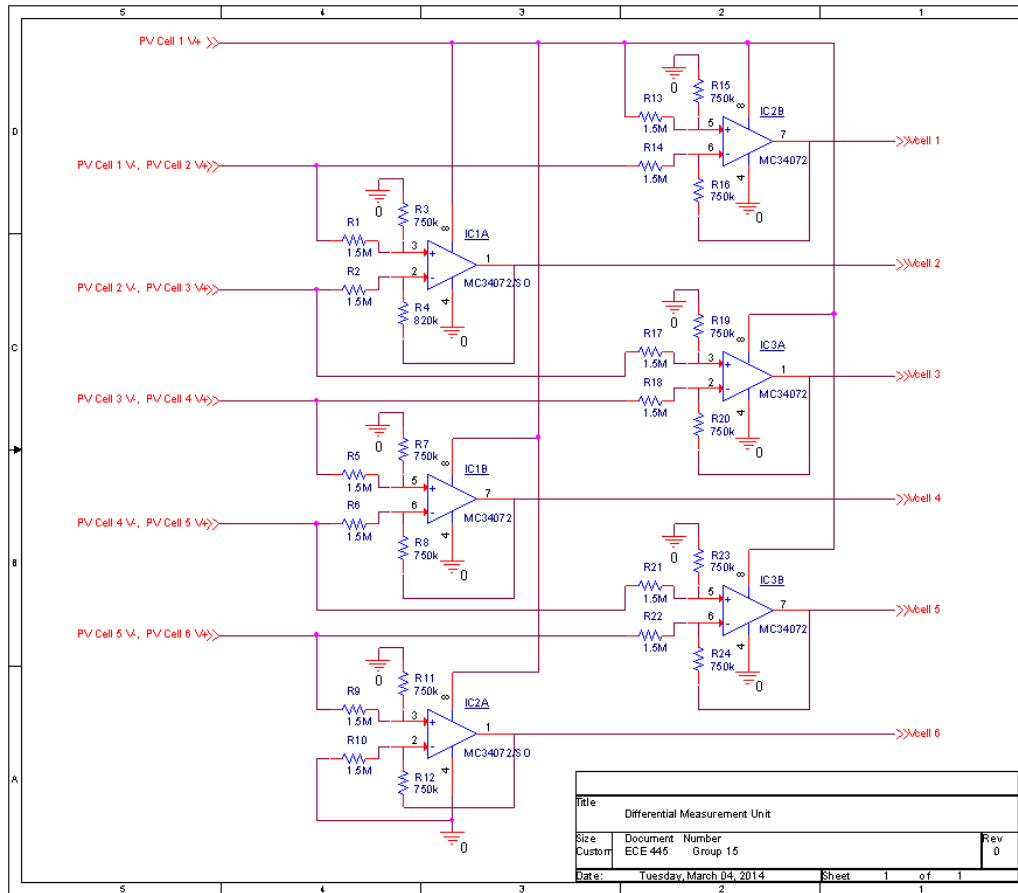


Figure 4. Differential Measurement Unit Schematic

2.4 Power Supply

2.4.1 Battery Charger

The battery is charged at 15V. The charger takes a 30V nominal input from the solar panel. Building a buck converter is the best fit for this situation because the output of the charger has lower voltage than the input. The input of the charger is limited to greater than 28V otherwise the converter shuts down from the microcontroller. This is because at approximately 28V, the solar panel voltage drops out. The switching frequency is selected to be 50kHz as supplied by the chosen microcontroller. Equation 5 describes the buck converter duty ratio. An important aspect of selecting the inductor is its current rating and its size. The current rating of needs to be able to handle the output current limit. If the input current drawn from the panels is around 0.3A, then the inductor current rating needs to be at least 0.6A. The size of the inductor is also important because it smooth out the output current spikes and make sure the current does not go towards discontinuous mode. The critical inductance is shown in Equation 6. The output capacitor is important to the design because it controls the output voltage spike. The output capacitor is determined by Equation 7 [4].

$$L_{crit} = \frac{D(V_{IN} - V_{OUT})}{(\Delta I_L)f_{SW}} = 294\mu H \quad (6)$$

$$C_{OUT} = \frac{1}{2} \frac{1}{2} \frac{2I_{OUT}}{f_{SW}} = 5.1 \mu F \quad (7)$$

The most important aspect of the converter is driving the switches with the required gate voltage. A gate driver is selected to boost up the voltage from a signal of 3.3V from the microcontroller to 12V from the battery. The MOSFET would need a high enough gate to source voltage in order to be on.

2.4.2 Battery

2.4.3 Logic Power Supply

6

microcontroller, when considered with the 1% measurement tolerance of the DMU, would not erroneously show a 10% drop in a PV Cell voltage. The +5V regulator did not need to have as tight of a tolerance. Therefore, the 5% tolerance of the +5V regulator was acceptable. The input and output capacitors for both regulators were selected based on datasheet recommendations [6] [7]. See Figure 6. Logic Power Supply Schematic for the circuit level schematic.

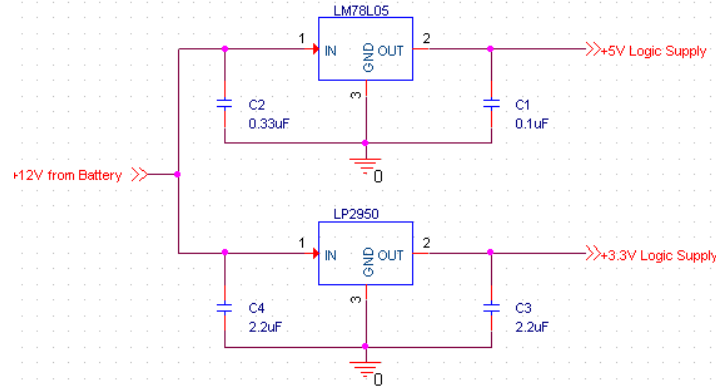


Figure 6. Logic Power Supply Schematic

2.4.4 Motor Power Supply

The MPS is a switched mode DC/DC buck converter that steps a nominal 12VDC down from the Battery to the nominal 7.2VDC motor voltage. Since this converter did not need to be controlled by the microcontroller a buck regulator IC was chosen with an integrated power MOSFET to simply design. This IC does have an enable that allows the microcontroller to turn off the MPS when not needed to power the DC motor.

It was expected that on average the motor could draw as much as 1.5A load based on a no load current of approximately 0.5A and the wiper weight being negligible. The MPS component values were calculated using switched-mode power supply analysis methods for an asynchronous buck converter [4]. The inductor current waveform was chosen to be on the verge of Discontinuous Mode to reduce its size with an average load current of 1.5A. The input voltage ripple was chosen to be less than 0.2V to keep the input voltage one order of magnitude below the expected Battery voltage range of 12.38V±7%. Output voltage was chosen to be less than 0.02V to be one order of magnitude smaller than the average output voltage specification. C_b was chosen as recommended by the datasheet [8]. The initial calculated values of the L , C_{IN} , and C_{OUT} are shown in Equations 8, 9, and 10 respectively. Except for the inductor L , actual parts used in the circuit may have higher component values due to cost and ratings. The inductor L value was increased to reduce output current spikes which would trigger the LM2678 current limiting function. The output voltage was set be regulated according to Equation 11 given by the LM2678 datasheet [8]. The R_2 value was made by 2 resistors in parallel. See Figure 7 for the circuit level schematic.

$$L = \frac{V_{out}(1 - D)}{f_{sw}\Delta i_L} = \frac{(7.2V)(0.4)}{(260kHz)(3A)} = 3.491\mu H, \quad \boxed{\text{Chose } L = 15\mu H} \quad (8)$$

$$C_{out} = \frac{0.5(DT)I_{max}}{\Delta V_{out}} = \frac{0.5(0.6)}{(260kHz)(0.02V)} = 81.82\mu F, \quad \boxed{\text{Chose } C_{out} = 470\mu F} \quad (9)$$

$$C_{in} = \frac{\Delta Q}{\Delta V_{in}} = \frac{(0.9A)(0.4)}{(260kHz)(0.2V)} = 6.545\mu F, \quad \boxed{\text{Chose } C_{in} = 1000\mu F} \quad (10)$$

$$R_2 = R_1 \left(\frac{V_{out}}{1.21} - 1 \right) = 1k\Omega \left(\frac{7.2V}{1.21} - 1 \right) = 4.95k\Omega, \quad \boxed{\text{Chose } 2 \parallel R = 10k\Omega} \quad (11)$$

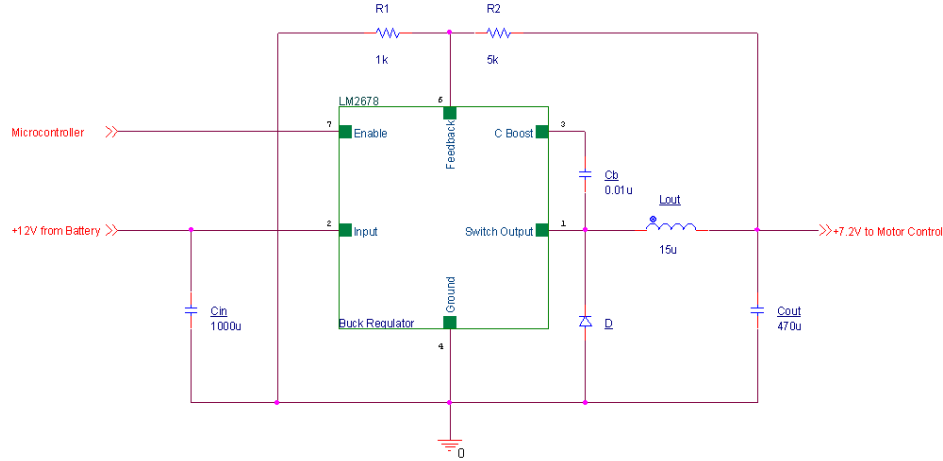


Figure 7. Motor Power Supply Schematic

2.5 Motor and Valve Control

2.5.1 Motor Control

The DC motor needs to have its own separate control circuit. The main component of the control circuit is a L298N H-Bridge Driver. The L298N is a high voltage, high current dual full-bridge driver. The full-bridge will be required to switch the direction of motor rotation with input combinations according to Table 1. The L298N IC needs to be supplied with $5V \pm 5\%$ for logic supply and $7.2V \pm 0.1V$ for the motor supply as shown in Appendix A. Both these supplies are designed separately.

We initially designed the motor control circuit to include a rotary encoder. The rotary encoder serves the purpose of signaling to the microcontroller that the direction of the motor needs to be changed. This is done by counting the number of rotations the rotary encoder goes through. After a certain number of rotations, which signals the end of the panel, the rotary encoder sends a signal to the microcontroller that the direction of the motor needs to be changed.

Fly-back diodes are used to prevent a strong electrical pulse from being sent through the circuit. The enable pin and the input pins come in from the microcontroller.

Table 1. Combinations of Inputs for Motor Control [9]

Inputs		Function
Ven = H	IN1 = H, IN2 = L	Forward
	IN1 = L, IN2 = H	Reverse
	IN1 = IN2	Fast Motor Stop
Ven = L	IN1 = X ; IN2 = X	Free Running Motor Stop

We started off with the initial design that we had proposed in the design review. The L298N IC was set up exactly as proposed and the power supplies, that we designed, successfully provided the logic power and motor power requirements.

However, when we were about to design the setup for the rotary encoder, Scott from the ECE Service Shop advised us to use two mechanical switches at each end of the panel to determine the position of the wiper. This idea appealed to us as it was simple to implement, and much more reliable.

Figure 8 and Figure 9 show the position of the switches in normal operation and when pressed, respectively. The *Common* terminal is connected to a 3.3V power supply from the microcontroller. When the switch is pressed against by the wiper, the voltage goes through the *Normally Open* terminal, which is connected as an input to the microcontroller. This signals the microcontroller to change the direction of the motor direction. The final implemented design of the motor control circuit is shown in Figure 10.

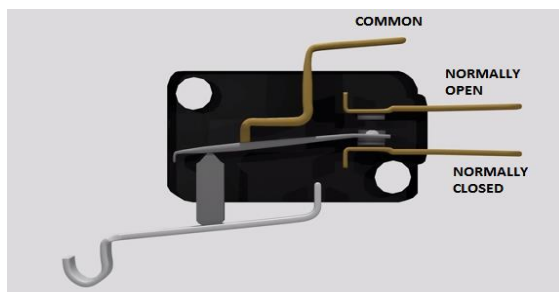


Figure 8. Switch in Normal Operation

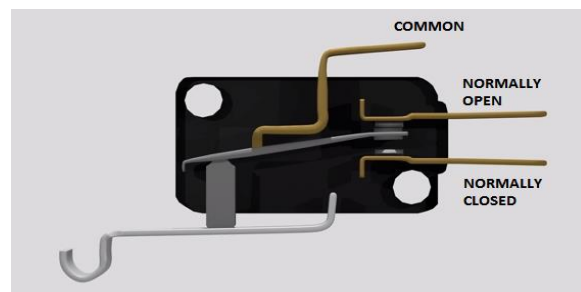


Figure 9. Switch when Pressed [10]

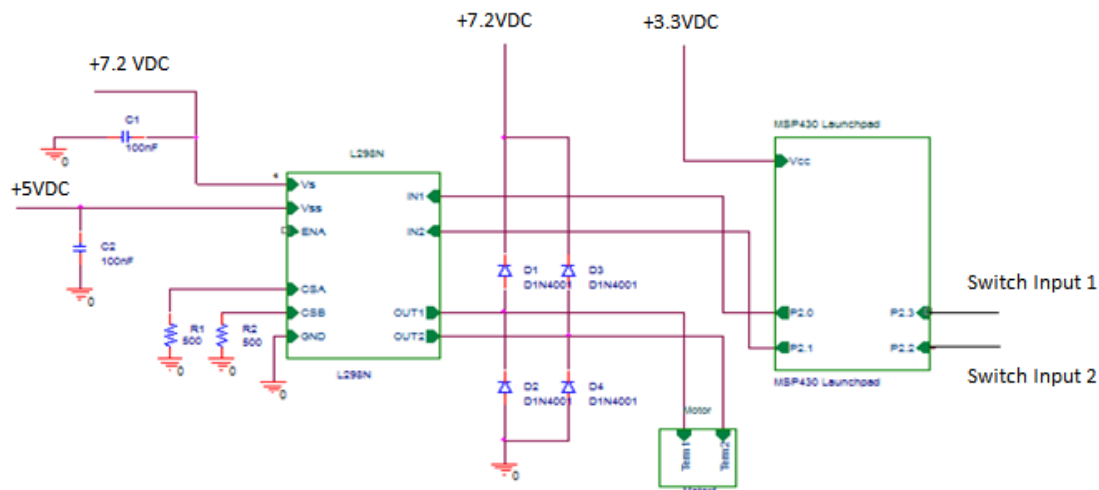


Figure 10. Final Design Schematic of Motor Control [11]

2.5.2 Valve Control

A solenoid valve is used to control when water is sprayed onto the panel. It is actuated at 12VDC, but according to the distributor, can also work down till 6VDC too [12]. Figure 11 shows how the valve will be connected into the circuit. It will be supplied with a voltage of 12VDC \pm 10%. The N-channel MOSFET is connected between the valve and the ground and will work as a control to open and close the valve. Whenever the N-Channel MOSFET gets a high input from the microcontroller, greater than 3VDC, it will cause the valve to open. The valve will only be opened for about 2-3 seconds before being closed again. This timing of the duration will be done through the microcontroller.

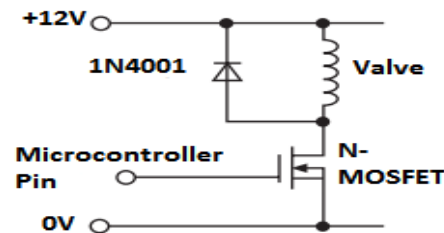


Figure 11. Solenoid Valve Circuit [13]

We implemented the valve control circuit exactly as we had initially proposed. We tested it with a power supply supplying the required voltages of 12V and 3.3V, and it seemed to actuate properly. However, when we tested it by powering the N-Channel MOSFET through a microcontroller, the solenoid valve didn't actuate. We later realized that the gate current provided by the microcontroller was insufficient to actuate the N-Channel Power MOSFET. We needed to insert a gate driver to solve this problem. But this issue was discovered after our demo, so we will keep it in mind for future considerations.

2.6 Cleaning Mechanism

This block of the project takes care of the required cleaning functions. The sprayer sprays water onto the panel, and the wiper mechanism wipes the panel.

2.6.1 Wiper

The wiper mechanism, which wipes the PV Panel in a linear motion, was fabricated by the ECE Machine Shop. The wiper mechanism consists of a linear actuator, along the center of the panel frame, driven by a DC motor. One wiper bar is connected at the actuator and stretches on both sides of the PV Panel. It is driven by the threaded rod's rotation.

2.6.2 Sprayer

The spray mechanism consists of pipes stretched along both sides of the panel. A sprayer nozzle for each solar cell is present on the pipe. This nozzle has an optimal spray angle. The solenoid valve controls the flow of water through the pipes. The source of water is a pressurized public water supply connected with a hose. This was fabricated by ECE Machine Shop.

3. Design Verification

3.1 Microcontroller

We observed all the actions happening according to Figure 2. All requirements are met on Table 13 with exception to the encoders.

3.2 PV Panel

Load testing was performed on the PV Panel to test for the requirements of the Design Review. The test was performed according to the Design Review Verification with two exceptions. First, the sunlight was artificially simulated in the PV Panel using a PV emulation technique [14]. Second, an electronic load was used to find the I-V curve of the PV Panel, which includes the open circuit voltage. The short circuit current was based on a single PV cell held directly up to a spot light: $I_{SC} = 0.9A$. The test results are shown graphically in Figure 12 and numerically in Table 2. All Design Review requirements were met. The Design Review requirements and verification for the PV Panel are listed in Table 14 in Appendix A.

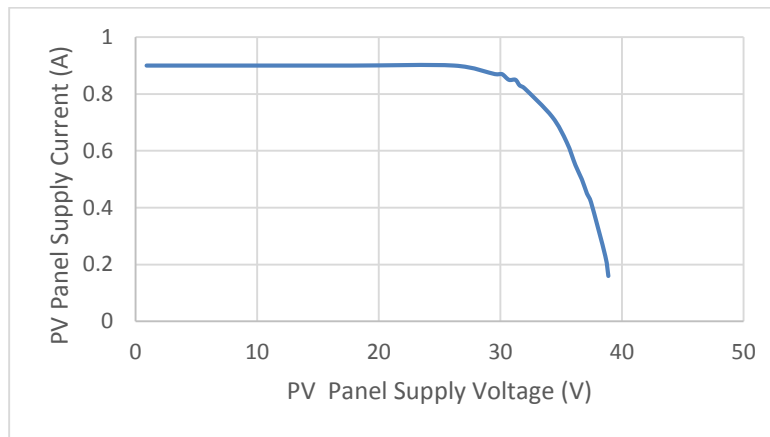


Figure 12. PV Panel I-V Characteristic Curve

Table 2. PV Panel Test Results

R_{LOAD} (Ω)	1	10	20	30	35	36	37	38	39	40	50	60	70	80	90	100	200	300
V_{LOAD} (V)	0.92	9.02	17.8	26.2	29.6	30.1	30.7	31.2	31.6	32.0	34.3	35.5	36.2	36.7	37.1	37.5	38.6	38.9
I_{LOAD} (A)	0.90	0.90	0.90	0.90	0.87	0.87	0.85	0.85	0.83	0.82	0.72	0.63	0.55	0.50	0.45	0.42	0.23	0.16

3.3 Differential Measurement Unit

The testing done on the DMU was different from the method specified in the Design Review. First, since all of the difference amplifiers are identical, only one sample needed to be tested to verify that it meets the tolerance requirement. Second, in order to fully test the difference amplifier it would be tested in

the two extreme cases: the amplifier at the highest PV Panel potential and amplifier is at the lowest PV Panel potential. In the case of the cell at the highest potential, the non-inverting terminal is connected to the highest PV Panel potential (V_{CC}). In the case of the cell at the lowest potential, the inverting terminal is connected to the lowest PV Panel potential (ground).

The test results are shown graphically in Figure 13 and Figure 14, and numerically in Table 3 and Table 4. All Design Review requirements were met as stated in Table 15 in Appendix A.

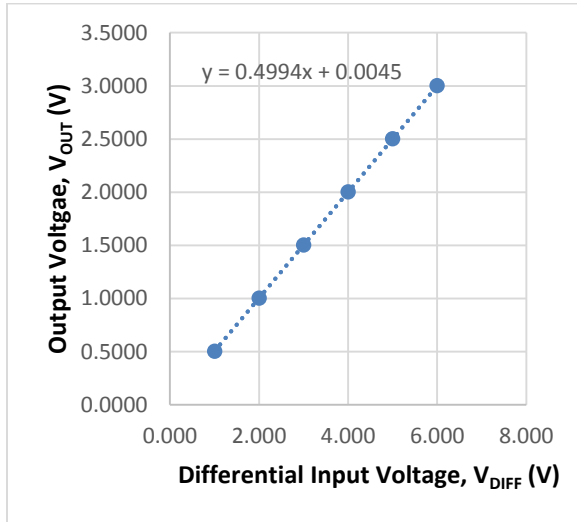


Figure 13. Output/Input Ratio for a Diff. Amp. At the "Lowest Potential PV Cell" position

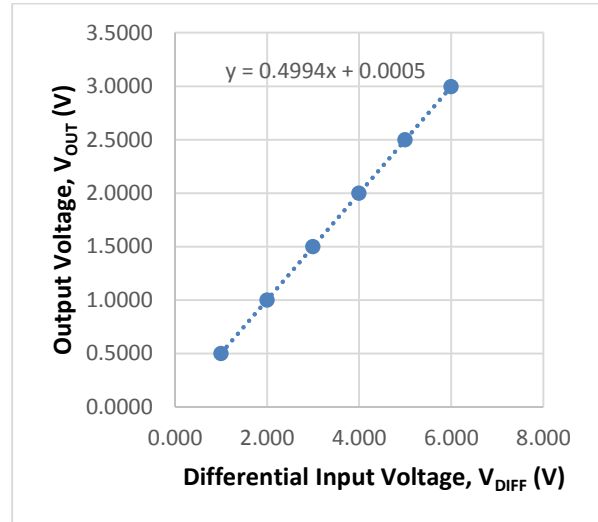


Figure 14. Output/Input Ratio for a Diff. Amp. At the "Highest Potential PV Cell" position

Table 3. "Lowest Potential PV Cell" Position Test Results

V_{DIFF} (V)	V_{ABOVE} (V)	V_{OUT} (V)
1.000	30.000	0.5037
2.000	30.000	1.0034
3.000	30.000	1.5030
4.000	30.000	2.0022
5.000	30.000	2.5016
6.000	30.000	3.0009

Table 4. "Highest Potential PV Cell" Position Test Results

V_{DIFF} (V)	V_{BELOW} (V)	V_{OUT} (V)
1.000	30.000	0.4998
2.000	30.000	0.9995
3.000	30.000	1.4988
4.000	30.000	1.9983
5.000	30.000	2.4977
6.000	30.000	2.9969

3.4 Power Supply

3.4.1 Battery Charger

The buck converter is tested with 30V from the Agilent E3631A power supply. The 50% PWM signal is generated by the MSP430G2553. To see the actual output voltage, a 15 Ω power resistor is used at the output replacing the battery. Figure 15 verified that the input and output voltages 30V and 15V, respectively. Figure 16 proves that the inductor current is in Continuous Current Mode since the output current is never zero. Figure 17 shows the output duty ratio of the gate drivers to be around 50%. At the

final design, we monitor the output voltage only to cut the buck converter circuit off when the output voltage is higher than 13V. Therefore, duty ratio is set at constant 50% and no longer varied.

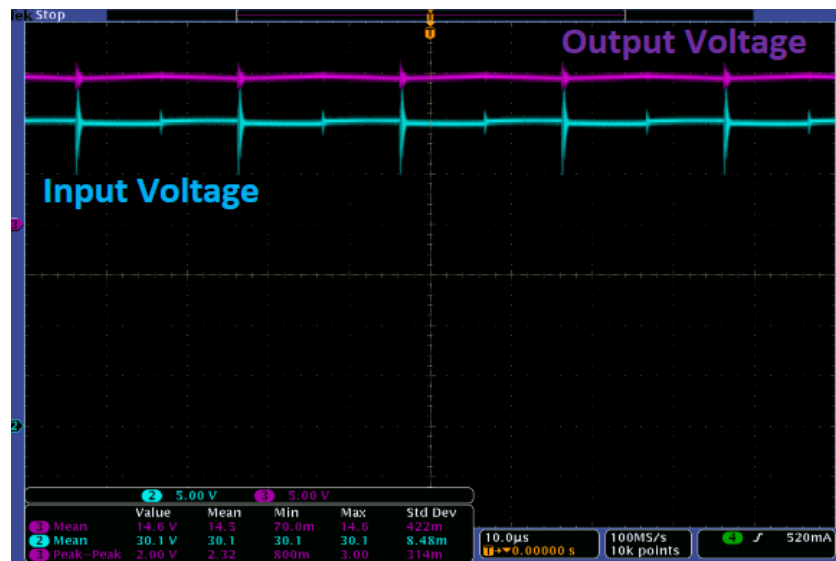


Figure 15. Input and Output Voltages of Buck Converter

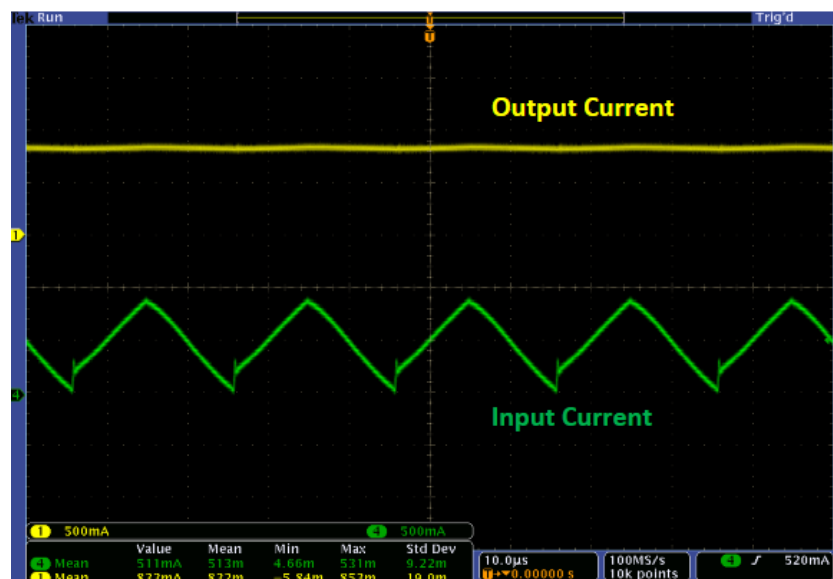


Figure 16. Input and Output Current of Buck Converter

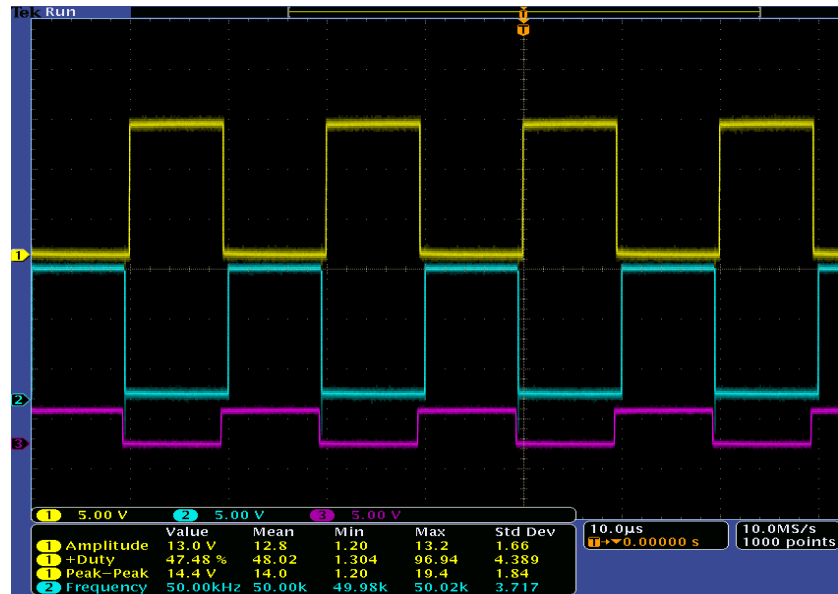


Figure 17. Synchronous 50% duty ratio of gate driver

3.4.2 Logic Power Supply

Both 3.3V and 5V power supplies were connected as in Figure 6. The input was connected to a bench top DC power supply with a corresponding 50mA output load for each regulator. Table 5 and Table 6 shows the results of the test over the expected range of Battery voltages. The +3.3V Supply was within the 1% tolerance for all output voltages as required by in Table 17 in Appendix A. Also, the +5V Supply was within the 5% tolerance for all output voltages as required by Table 17 in Appendix A.

Table 5. +5V Logic Power Supply Test Results

V _{IN} (V)	V _{OUT} (V)	V _{OUT,IDEAL} (V)	%Error (V _{OUT} -> V _{OUT,IDEAL})
11.50	5.0557	5	1.11%
11.75	5.0557	5	1.11%
12.00	5.0556	5	1.11%
12.25	5.0554	5	1.11%
12.50	5.0553	5	1.11%
12.75	5.0551	5	1.10%
13.00	5.0549	5	1.10%
13.25	5.0548	5	1.10%
AVERAGE	5.0553125	5	1.11%

Table 6. +3.3V Logic Power Supply Test Results

V _{IN} (V)	V _{OUT} (V)	V _{OUT,IDEAL} (V)	%Error (V _{OUT} -> V _{OUT,IDEAL})
11.50	3.2840	3.3	0.48%
11.75	3.2840	3.3	0.48%
12.00	3.2848	3.3	0.46%
12.25	3.2838	3.3	0.49%
12.50	3.2840	3.3	0.48%
12.75	3.2841	3.3	0.48%
13.00	3.2841	3.3	0.48%
13.25	3.2844	3.3	0.47%
AVERAGE	3.28415	3.3	0.48%

3.4.3 Motor Power Supply

The MPS was tested according to the Design Review verification and met the Design Review requirement shown in Table 18 in Appendix A. Test results are listed in Table 7. Though the MPS tested satisfactory with the planned test procedure, the operational test with the DC motor driving the wiper blade on the PV Panel initially failed. The failure was due to the current limiting feature of the LM2678 IC causing the output voltage to sag near 2V with no motor motion. Current surges exceeding the rated

5A of the LM2678 IC were caused by unexpected friction in the threaded rod driving the wiper blade. Grease applied to the rod reduced friction and improved the output voltage to approximately 4V which was capable of driving the DC motor.

Table 7. Motor Power Supply Test Results

V_{IN} (V)	V_{OUT} (V)
11.50	7.184
11.75	7.183
12.00	7.181
12.25	7.179
12.50	7.177
12.75	7.176
13.00	7.175
13.25	7.174

3.5 Motor and Valve Control

3.5.1 Motor Control

In order to test whether the motor control circuit was being supplied with the correct voltages, we connected the motor and logic power supplies, along with the pins from the microcontroller. We measured the voltages using a voltmeter. We tried different combinations of the input pins on the L298N IC and verified changes in motor direction.

We connected an LED between the *Normally Open* terminal of the switch and ground and verified that pressing against the switch lights up the LED. This was done before mounting the switches onto the panel.

3.5.2 Valve Control

We set up the circuit according to Figure 11. We applied a voltage at the gate of the N-Channel MOSFET using a power supply, and verified that the valve actuates by hearing for a ‘click’ inside the solenoid valve. Table 8 shows the operating voltages of the solenoid valve. The lower the voltage is, the slower the valve actuates. This process, however, came to bite us back in the end since we had overlooked to test whether there is sufficient gate current being supplied from the microcontroller.

Table 8. Operating Voltages of Solenoid Valve

Voltage (V)	Current (A)
12.00	0.456
11.00	0.411
10.00	0.372
9.00	0.332
8.00	0.295
7.00	0.258
6.00	0.222

3.6 Cleaning Mechanism

3.6.1 Wiper

Using a battery, we applied 12V at the motor connected to the panel to verify the wiper movement across the panel.

3.6.2 Sprayer

We could not test the valve with water in the lab. However, after the demo, we went outside and connected a water supply to the panel using a garden hose. Then, using a battery, we applied 12V at the solenoid valve. This allowed water to be sprayed onto the panel.

4. Costs

4.1 Parts

Table 9. Parts List

Circuit	Part	Manufacturer	Quantity	Retail Cost (\$/unit)	Actual Cost (\$)
PV Panel	PV Cell	Not Known (Amazon)	6	8.00	48.00
LPS	+5V Linear Reg. (LM78L05)	Texas Instruments	1	0.44	0.44
LPS	+3.3V Linear Reg. (LP2950-3.3)	Texas Instruments	1	0.36	0.36
LPS	0.33 μ F Capacitor (UVR1HR33MDD1TD)	Nichicon	1	0.14	0.14
LPS	0.1 μ F Capacitor (UVR2A0R1MDD1TD)	Nichicon	1	0.14	0.14
LPS	2.2 μ F Capacitor (UVK1H2R2MDD1TD)	Nichicon	1	0.14	0.14
MPS	Buck Reg. IC (LM2678)	Texas Instruments	1	3.62	3.62
MPS	Schottky Diode (1N5819)	International Rectifier	1	0.55	0.55
MPS	10nF Capacitor (AR155C103K4R)	AVX	1	0.43	0.43
MPS	1000 μ F Capacitor (EEU-FR1V102B)	Panasonic	1	1.32	1.32
MPS	470 μ F Cap. (16ZLH470MEFC)	Rubicon	1	0.40	0.40
MPS	1k Ω Resistor (MFR-25FBF52-1K)	Yageo	1	0.10	0.10
MPS	10k Ω Resistor (MFR-25FBF52-10K)	Yageo	2	0.10	0.10
DMU	Dual Opamp IC (MC34072PGOS-ND)	ON Semiconductor	3	1.05	3.15
DMU	1.5M Ω Resistor (1-1614959-1)	TE Connectivity	12	0.65	7.80
DMU	750k Ω Resistor (ERA-3AEB751V)	Panasonic	12	0.43	5.16
Battery Charger	Npn-MOSFET (512-RFP50N06)	Fairchild Semiconductor	2	1.50	3.00
Battery Charger	Gate Driver (IR2302PBF)	Mouser Electronics, Inc	1	3.33	3.33
Battery Charger	33 Ω Resistor (RNMF14FTC33R0)	Stackpole Electronics Inc	2	.009	.018
Battery Charger	820 μ H Inductor (RFS1317-824KL)	Coilcraft	1	2.50	2.50
Battery Charger	15 μ F Capacitor (FK24X5R0J156M)	TDK Corporation	1	0.45	0.45
Battery Charger	1 μ F Capacitor (K105K20X7RF5UH5)	Vishay BC Components	2	0.19	0.38
Battery Charger	30 μ F Capacitor (FK16X5R0J336M)	TDK Corporation	1	0.71	0.71
Battery	12V Lead Acid Battery (UB1270)	UPG	1	14.00	14.00

Battery Charger	Diode (1N4002)	Fairchild Semiconductor	1	0.13	0.13
Microcontroller	Microcontroller (MSP430G2553)	Texas Instruments	1	13.00	13.00
Motor Control Unit	Dual Full-Bridge (L298N)	STMicroelectronics	1	2.95	2.95
Sprayer Unit	Plastic Water Solenoid Valve (N/A (Adafruit))	Adafruit	1	6.95	6.95
Sprayer Unit	N Channel MOSFET (MTP36N06V)	ON Semiconductor	1	Free	Free
Motor Control, Sprayer	Diode Rectifiers (1N4001)	International Rectifier	5	0.15	2.25
Motor Control	100nF		2	0.26	1.3
Motor Control	Mechanical Micro-switches (BZ-2RW822-A2)	Honeywell	2	12.76	25.52
Motor Control	Motor (MP-36020-385)	Banebots	1	Free	Free
Perfboard		Radioshack	4	4.00	16.00
TOTAL PARTS COST					164.34

4.2 Labor

Table 10. Labor Costs

Name	Hourly Rate (\$/hr)	Total Hours Invested (hr)	Total (\$) = Hourly Rate x 2.5 x Total Hours Invested
ECE Machine Shop	20.00	10	200.00
Yann-Tyng Lin	50.00	150	18750.00
Yousaf Abdul Salam	35.00	150	13125.00
Terry Green	35.00	150	13125.00
Total	120.00	450	45200.00

5. Conclusion

5.1 Accomplishments

The fully assembled system was able to detect a shaded cell from debris. Furthermore, it initiated the wiper motion down and up the panel to clear the debris. Also, the system maintained the battery charged when there was no cleaning and sufficient power was available.

More importantly, the project decreased the daily energy lost compared to the case where the PV panel was left shaded for an entire day. In order to determine energy savings, the PV Panel was placed under a solar test bed under identical conditions to test the energy loss from one half shaded g a cell. The losses due to shading alone was determined by taking the difference between the maximum power of the unshaded PV Panel and the maximum power of with one PV cell half shaded. The power losses of the entire Automated Self-Cleaning Solar Panel was also measured. Energy loss was normalized for one day of operation with one cleaning cycle. The results are tabulated in Table 11 and Table 12. The resulting percent energy savings is shown in Equation 12.

$$\%Energy\ Saved = \frac{Total\ Energy\ Saved}{Total\ Shading\ Energy\ Loss} = \boxed{39.9\%} \quad (12)$$

Table 11. Energy Losses Per Day Due to One Half Shaded Cell

	Electronic Load Power (W)	Electronic Load Current (A)	Time (sec)	Energy (Ws)
Unshaded Panel	4.8	0.13	86400	414720
One Cell Half Shaded	2.7	0.074	86400	233280
ENERGY LOSS DUE TO SHADING				181440

Table 12. Circuit Energy Losses Per Day

	Current (A)	Voltage (V)	Average Power (W)	Run Time (sec)	Waste Energy (Ws)
Motor	1.2	12.2	14.64	8	117.12
Solenoid Valve	0.456	12	5.472	5	27.36
Differential Measurement Unit	0.037	30	1.11	86400	95904
Other Continuous Running Circuits	12.5	0.012	0.15	86392	12958.8
TOTAL ENERGY LOSS					109007

5.2 Uncertainties

Although our project worked almost exactly as we had planned, and each group member was satisfied with our final product, there were still some uncertainties that we could not address during the duration of this course.

One uncertainty was whether we could make a significant long-term profit. We were able to perform preliminary tests and calculations to determine our daily, or even weekly, profit from the design. But we could not extrapolate that data to estimate our profit on a monthly or yearly basis because of the possibility of fluctuations in weather and long-term maintenance costs.

Another concern we had was marketability. What areas would our project be beneficial for? Would it be more beneficial on a residential scale, or an industrial scale? All these questions need to be addressed before our product can be produced on a large scale.

5.3 Ethical considerations

The following is a list IEEE Codes of Ethics that were most related to the project [15]. The original Codes are in italics with an explanation as to their relevance to the project.

1. To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment

We have made a list of safety precautions during the design review phase to prevent the injury to anyone that may come in contact with a working version of our project.

3. To be honest and realistic in stating claims or estimates based on available data; to reject bribery in all its forms

Although, bribery hasn't been an issue, it's worth noting that the project's performance is measured according to data collected from real tests. That way there is no surprise as to the behavior of the project if someone else were to pick it up if we were to turn it over to another team or a consumer.

4. To improve the understanding of technology; its appropriate application, and potential consequences

Throughout the design and testing our group is gaining a more in depth feel for how electrical engineering is actually applied to create a real system that has a definite purpose in the real world. Also, the consequences of poor design are much more evident.

5.4 Future work

Even though our project worked perfectly and was functioning as initially planned, there are still a lot of improvements that can be made to make it more marketable and efficient.

- First of all, we would like to deal with reducing the friction losses in our project. It is something we did not account for since our panel was prepared by the ECE Machine Shop. We could use a better, and more lubricated ball-screw which would significantly reduce our friction losses.

- We would also work on making our project more marketable by designing and ordering a PCB. This would make our circuit look neat, and would also reduce its size considerably.
- We would like to make our DMU more universal. We could make our DMU wireless, which would allow us to relay data back to a central monitoring system.
- Research public opinion and determine whether people would be willing to pay for the long-term savings that our project promises.

References

- [1] G. Masters, Renewable and Efficient Electric Power Systems, Hoboken, NJ: John Wiley & Sons Inc., 2013.
- [2] A. S. Sedra and K. C. Smith, Microelectronic Circuits, 6th ed., New York, New York: Oxford University Press, 2010.
- [3] Industries, Semiconductor Components, datasheet, *MC34072*, 21st ed., 2013.
- [4] P. T. Krein, Elements of Power Electronics, 1st ed., New York: Oxford University Press, 1998.
- [5] *Sealed Lead Acid Battery*, datasheet, UPG. Available at: <http://powerupco.com/site/wp-content/files/Batteries/UPG/UB1270.pdf>.
- [6] Texas Instruments Incorporated, datasheet, *LM78LXX Series 3-Terminal Positive Regulators*, 2013.
- [7] Texas Instruments Incorporated, datasheet, *LP2950 Series of Adjustable Micropower Voltage Regulators*, 2013.
- [8] Texas Instruments Incorporated, datasheet, *LM2678 SIMPLE SWITCHER High Efficiency 5A Step-Down Voltage Regulator*, 2013.
- [9] *L298 Dual Full-Bridge Driver*, datasheet, STMicroelectronics, 2000.
Available at: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf.
- [10] *Understanding a Microswitch*. Youtube. [Online]. Available: <https://www.youtube.com/watch?v=q6nP1FjxAMU>.
- [11] *Controlling motors with an MSP430 (or Arduino)*. Maker Dreams. [Online]. Available: <http://lars.roland.bz/controlling-motors-with-an-msp430/>.
- [12] *Plastic Water Solenoid Valve*. Adafruit. [Online]. Available: <http://www.adafruit.com/products/997#Description>.
- [13] *Microcontroller Interfacing Circuits, 2000*. [Online]. Available: http://faculty.unlv.edu/pushkin/microcontroller_interfacing_revolution.pdf.
- [14] Qin, Shibin; Kim, K.A.; Pilawa-Podgurski, R.C.N., "Laboratory emulation of a photovoltaic module for controllable insolation and realistic dynamic performance," Power and Energy Conference at Illinois (PECI), 2013 IEEE , vol., no., pp.23,29, 22-23 Feb. 20.
- [15] "IEEE Code of Ethics," [Online]. Available: <http://www.ieee.org/about/corporate/governance/p7-8.html>. [Accessed 18 March 2014].

Appendix A Requirement and Verification Table

Table 13. Microcontroller Requirements and Verifications

Verification status (Y or N)	Requirement	Verification
Y	Microcontroller must be able to interpret the signals from the six differential probes and the two encoder ports, then command the action of the motor, the Battery Charger, and the solenoid valve.	Place a voltmeter each at the terminal of the valve, the gate driver, two terminals at the encoder, 6 terminals at the differential probes, and three terminals at the H-bridge to see the voltage values. Check to see if the voltage values match the logic below.
Y	The solenoid valve enable pin and the H-bridge enable pin from the microcontroller must output a 3.0V signal if any of the 6 differential probes sense less than 90% of the averaged voltage in 1 minute. The pins must output 0.3V when all the differential probes sense higher than 90% of the averaged voltage in 1 minute. The gate driver PWM will output 0.3V when the other two enable pins are 3.0V and output 3.0V when the other two enable pins are 0.3V.	With illumination over all six panels, cover half of one panel up for 1 minute and place a voltmeter on the solenoid valve enable pin and the H-bridge enable pin. Check to see if the meters read 3.0V. Conversely, leave all panels unblocked for 1 minute and make sure the voltmeters are reading 0.3V at the solenoid valve enable pin and the H-bridge enable pin. Check the gate driver pin to see if the signal is having the opposite action.
N	The microcontroller must register 1 count when either of the encoder pins signals change values from 3.0V to 0V or vice versa.	Create a watch expression on the count based on the change of any encoder bits. Verify on the computer screen that the count increments.
N	At count 3072, the microcontroller must output the signal to reverse the H-bridge direction and reset the count.	As the computer shows 3072, measure the 2 output pins to the H-bridge with a voltmeter. Make sure the pin signals switch from 0V to 3.0V and 3.0V to 0V.

Table 14. PV Panel Requirements and Verifications

Verification status (Y or N)	Requirement	Verification
Y	The PV Panel must supply sufficient open circuit DC voltage to the Battery Charger:	Supply solar test bed with bench DC supply.

	$V_{\text{PANEL}} \geq 15\text{V}$ when panel is provided with a light source to generate a short circuit current between 0.8A and 0.9A.	Adjust the DC supply to achieve PV Panel short circuit current between 0.8A and 0.9.
		Use a digital multimeter to measure the PV panel open circuit voltage.
		The PV panel open circuit voltage must be greater than or equal to 15V.

Table 15. Differential Measurement Unit Requirements and Verifications

Verification status (Y or N)	Requirement	Verification
Y	Each translated differential voltage must be no greater than 3.3VDC at the output of each of the six difference amplifiers for a differential input voltage of no more than $\pm 0.1\text{VDC}$.	Connect a bench top DC power supply to the differential inputs of any one of the difference amplifiers. The polarity of the connections between the power supply and the amplifier should match the polarity shown in Figure 17.
		Set the voltage of the bench top DC power supply between 5.9V and 6.0V.
		Measure the chosen difference amplifier output voltage.
		Repeat steps 1 through 3 for all six difference amplifiers.
		Each amplifier output should be less than 3.3VDC.
Y	The output of the difference amplifiers must scale down by half of the differential input within $\pm 1\%$. In other words, for each amplifier, the ratio of the amplifier output voltage to the input voltage must be 0.495 to 0.505. The requirement must hold for differential input voltages between $1.0 \pm 0.1\text{V}$ and $6.0 \pm 0.1\text{V}$.	Connect a bench top DC power supply to the differential inputs of any one of the difference amplifiers. The polarity of the connections between the power supply and the amplifier should match the polarity shown in Figure 17.
		Set the voltage of the bench top DC power supply to $1.0 \pm 0.1\text{V}$
		Note the voltage as read on the bench top DC power supply.
		Measure the chosen difference amplifier

		output voltage.
		Repeat steps 2 through 4 for bench top voltages $1.5\pm0.1V$, $2.0\pm0.1V$... $6.0\pm0.1V$.
		Repeat steps 2 through 5 for all six amplifiers.
		For each amplifier, the ratio of the amplifier output voltage to the input voltage must be 0.495 to 0.505.

Table 16. Battery Charger Requirements and Verifications

Verification status (Y or N)	Requirement	Verification
Y	Battery must be charged between 14.4V - 15.0V	Use an oscilloscope and a voltmeter to monitor the output capacitor voltage and make sure the peak-to peak voltage is within 14.4V to 15.0V.
Y	Power supply is used for the input voltage and current, and it must provide between 30V-15V, 0.5A.	Use a voltmeter, a current meter, and an oscilloscope to verify the input voltage and current are between 30V-15V and 0.5A.
Y	MOSFET driven by the gate driver and a waveform generator is producing the correct duty ratio.	Make sure the duty ratio of the waveform generator is set to square wave with the correct duty ratio of $D= V_{out}/V_{in}$. Measure the output waveform of the MOSFET with an oscilloscope to see if the correct duty ratio is displayed. The correct duty ratio is defined to range from 0.49 to 0.98.
Y	Inductor must operate at Continuous Conduction Mode. (CCM)	Use a current probe and make sure current across the inductor does not go below 0A.
Y	Closed loop design must replace MSP430 as the duty ratio generator.	Before hooking the microcontroller up with the whole Battery Charger, probe the waveform out of the PWM terminal (P2.5) of the microcontroller with an oscilloscope and make sure the duty ratio is correct.

N	The MSP430 PWM responds to the change in output voltage.	Connect the ADC terminal (P1.6) of the MSP430 to the output node of the Battery Charger with the lab waveform generator. Use an oscilloscope to probe the PWM terminal (P2.5) and make sure the duty cycle increases as the output voltage is less than $V_{ref-min} = 14.4V$ and the duty cycle decreases as the output voltage is greater than $V_{ref-max} = 15.0V$. Probe the output terminal voltage, and make sure the duty ratio on the oscilloscope reaches stability if the output voltage is probed between 14.4V and 15.0V.
---	--	---

Table 17. Logic Power Supply Requirements and Verifications

Verification status (Y or N)	Requirement	Verification
Y	Provides 5 VDC $\pm 5\%$ at the output for a 100 Ω load. The range of input voltages is allowed to be between 11.5VDC and 13.25VDC.	Connect a bench top DC power supply to the input of the Logic Power Supply.
		Connect an electronic load to the output at the +5V Logic Supply.
		Set the electronic load to act as a 100 Ω load.
		Set the voltage of the bench top DC power supply to 11.0 $\pm 0.1V$
		Measure the Logic Power Supply output voltage.
		Repeat steps 2 through 4 for bench top voltages 11.5 $\pm 0.1V$, 11.75 $\pm 0.1V$... 13.25.0 $\pm 0.1V$.
		The Logic Power Supply output voltage at the +5V Logic Supply terminal must be 5 VDC $\pm 5\%$ for all input values tested.
Y	Provides 3.3 VDC $\pm 1\%$ at the output for a 100 Ω load. The range of input voltages is allowed to be between 11.5VDC and 13.25VDC.	Connect a bench top DC power supply to the input of the Logic Power Supply.
		Connect an electronic load to the output at the +3.3V Logic Supply.
		Set the electronic load to act as a 100 Ω load.
		Set the voltage of the bench top DC power supply to 11.0 $\pm 0.1V$
		Measure the Logic Power Supply output voltage.
		Repeat steps 2 through 4 for bench top

		voltages 11.5±0.1V, 11.75±0.1V... 13.25.0±0.1V.
		The Logic Power Supply output voltage at the +5V Logic Supply terminal must be 5 VDC ±5% for all input values tested.

Table 18. Motor Power Supply Requirements and Verifications

Verification status (Y or N)	Requirement	Verification
Y	Provides 7.2±0.1VDC at the output for a 4.8Ω load. The range of input voltages are allowed to be between 11.5VDC and 13.25VDC.	Connect a bench top DC power supply to the input of the Motor Power Supply.
		Connect an electronic load to the output of the Motor Power Supply.
		Set the electronic load to act as a 4.8Ω load.
		Set the voltage of the bench top DC power supply to 11.0 ±0.1V
		Measure the Motor Power Supply output voltage.
		Repeat steps 2 through 4 for bench top voltages 11.5±0.1V, 11.75±0.1V... 13.25.0±0.1V
		Connect a bench top DC power supply to the input of the Motor Power Supply.

Table 19. Motor Control Requirements and Verifications

Verification status (Y or N)	Requirement	Verification
	L298N IC	
Y	The logic blocks are receiving a supply voltage of 5VDC ± 5%	Use a voltmeter to check the voltage at the logical supply pin on the L298N IC and display output on an oscilloscope.
Y	The supply voltage that will be fed to motor is 7.2VDC ± 0.1V.	Use a voltmeter to check the voltage at the Vs pin on the L298N IC and display output on an oscilloscope.
Y	The motor enable pin is receiving a DC voltage ≥ 3V±5% to enable the H-bridge for motor	Use a voltmeter to check the voltage at the enable pin for Bridge A on the L298N IC and display output on an oscilloscope.
Y	The TTL compatible inputs of Bridge A switch the motor direction depending on their combination: IN1:IN2 = 1:0 – Forward	Connect a motor to the IC and verify the direction of motion for different combinations of the input pins.

	IN1:IN2 = 0:1 – Reverse IN1:IN2 = 0:0 or 1:1 – Stop	
	Rotary Encoder	
N/A (See Section. 2.5.1)	The Pins A and B are receiving a supply voltage of 5VDC \pm 5%	Use a voltmeter to check the voltage at the pins on the encoder and display output on an oscilloscope.
N/A (See Section. 2.5.1)	<p>Pins A and B make transitions from low to high, or vice versa, in the correct order depending on direction of rotation of the encoder:</p> <p>A \uparrow , B = 0V (cw), B \uparrow , A = 0V (ccw) B \uparrow , A = 5V (cw), A \uparrow , B = 5V (ccw) A \downarrow , B = 5V (cw), B \downarrow , A = 5V (ccw) B \downarrow , A = 0V (cw), A \downarrow , B = 0V (ccw)</p>	Connect an LED to each of Pin A and Pin B and rotate the encoder in both directions and verify that the order the LEDs turn on and off in is as per transitions of the encoder.
N/A (See Section. 2.5.1)	The encoder has 24 pulses per revolution	Using the same LEDs, ensure that any one of them turns on and off 24 times in a 360° rotation of the encoder knob. (This will be programmed later so that the counts are incremented on a microcontroller).

Table 20. Valve Control Requirements and Verifications

Verification status (Y or N)	Requirement	Verification
Y	The valve is receiving a supply voltage of 12VDC \pm 10%	Use a voltmeter to check the voltage and display output on an oscilloscope.
Y	The N-Channel MOSFET is receiving a Gate voltage of \geq 3VDC \pm 3%	Use a voltmeter to check the voltage and display output on an oscilloscope.
Y	The response time of the solenoid valve should be \leq 1s	Have an LED connected to the circuit to detect when the solenoid gets an electrical signal. Start a stopwatch and as soon as the solenoid opens stop the stopwatch. Since it'll be a really quick process repeat the procedure several time and take an average.

Appendix B MSP430 Code

```
#include <msp430.h>

/*****
#DEFINES
*****/
// TERRY
#define TOTAL_SAMPLES 10
#define TIMER0_INT0_PERIOD 15800 // ~1ms period measured by o-scope

// YOUSAF
#define P17 BIT7 // Pin input for switch at beginning of panel
#define P20 BIT0 // Pin output for solenoig valve
#define P22 BIT2 // Pin input from switch at end of panel
#define P23 0x08 // Pin output for motor enable
#define P24 0x10 // Pin output for motor direction
#define P25 0x20 // Pin output for motor direction

/*****
GLOBAL VARIABLES
*****/
volatile unsigned adc_sample[7];
volatile unsigned adc_arr[7][TOTAL_SAMPLES];
volatile unsigned sample_cnt = 0;
volatile unsigned millisCount = 0;
unsigned shadedTime0 = 0;
unsigned shadedTime1 = 0;
unsigned shadedTimeDiff = 0;
unsigned shadedTimeTotal = 0;

// Elmo
unsigned BattVoltage = 0; // 13V at battery is 2.206V at uC
unsigned PanelVoltage = 0;

/*****
FUNCTION DECLARATIONS
*****/

// TERRY
void setupSysClk(void);
void setup_ADC(void);
void setup_pins(void);
void setupTimer0(void);
void enableTimer0Int_0(void);
void Timer1PWM(void);
void read_analog(void);
unsigned single_cell_avg(int);
unsigned panel_avg_wrt(int,unsigned, unsigned, unsigned,
                      unsigned, unsigned, unsigned);
unsigned shaded_cell(void);
unsigned shaded_cell_delay(void);
```

```

// YOUSAF
void Motor_FWD(void); // allows motor to move in forward direction
void Motor_REV(void); // allows motor to move in reverse direction
void Motor_STOP(void); // stops motor
void MOTOR_CONTROL(void); // Controls movement of motor across panel
void VALVE_CONTROL(void); // Controls the valve opening/closing

/*****
MAIN CODE
*****/
void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT
    _BIS_SR(GIE); // Enable global interrupt
    setupSysClk();
    setup_ADC(); //setup the ADC
    setup_pins(); //setup IO
    setupTimer0();
    enableTimer0Int_0();

    Timer1PWM();

    while(1)
    {
        if(shaded_cell_delay())
        {
            P2OUT &= ~BIT6;
            P2OUT &= ~BIT7;
            if(BattVoltage < 697)
            {
                VALVE_CONTROL();
                MOTOR_CONTROL();
            }
        }
        else
        {
            P2OUT &= ~P23;

            BattVoltage = single_cell_avg(6);

            PanelVoltage = single_cell_avg(0) + single_cell_avg(1)
                          + single_cell_avg(2) + single_cell_avg(3)
                          + single_cell_avg(4) + single_cell_avg(5);

            if( (BattVoltage < 684) && (PanelVoltage > 4467) )
            {
                P2OUT |= BIT6;
                P2OUT &= ~BIT7;

                unsigned battcountms;
                for(battcountms = 0; battcountms < 1000; battcountms++)
                {
                    unsigned battcount;
                    for(battcount = 0; battcount < 16000; battcount++){

```

```

    }
    else
    {
        P2OUT |= BIT7;
        P2OUT &= ~BIT6;
        /*
        unsigned battcountms;
        for(battcountms = 0; battcountms < 5000; battcountms++)
        {
            unsigned battcount;
            for(battcount = 0; battcount < 16000; battcount++){
        }
        */
    }
}

}

}

}

/*****
    INTERRUPT SERVICE ROUTINES
    *****/
// ADC10 interrupt service routine
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
    adc_arr[0][sample_cnt] = adc_sample[6];
    adc_arr[1][sample_cnt] = adc_sample[5];
    adc_arr[2][sample_cnt] = adc_sample[4];
    adc_arr[3][sample_cnt] = adc_sample[3];
    adc_arr[4][sample_cnt] = adc_sample[2];
    adc_arr[5][sample_cnt] = adc_sample[1];
    adc_arr[6][sample_cnt] = adc_sample[0];
}

// Timer0 A0 interrupt service routine
#pragma vector = TIMER0_A0_VECTOR
__interrupt void Timer0_A0 (void)
{
    millisCount += 1;
    TACCR0 += TIMER0_INT0_PERIOD;
}

/*****
    FUNCTION DEFINITIONS
    *****/
void setupSysClk()
{
    DCOCTL = CALDCO_16MHZ;
    BCSCTL1 = CALBC1_16MHZ;
}

void setupTimer0()
{
    TACTL = TASSEL_2 | MC_2 | ID_0;
}

```

```

void enableTimer0Int_0()
{
    TACCTL0 = CCIE;
    TACCR0 = TIMER0_INT0_PERIOD;
}

//IO pins setup function
void setup_pins()
{
    // TERRY
    /*
     * P2.7 indicates idle condition on LED
     */
    P2SEL &= ~BIT7; // P2.7 not used at crystal osc
    P2DIR |= BIT7; // Set P2.7 to output
    P2OUT &= ~BIT7; // Clear P2.7

    // YOUSAF
    //P2DIR = 0x00; // Set all P2.X pins as inputs
    P1DIR &= ~P17; // Signal input from switch at top of panel
    P2DIR &= ~P22; // Signal input from switch at top of panel
    P2DIR |= P23; // Set as output to enable
    P2OUT &= ~P23;
    P2DIR |= P24; // Set as output to one input of H-bridge for motor
direction
    P2OUT &= ~P24;
    P2DIR |= P25; // Set as output to other input of H-bridge for motor
direction
    P2OUT &= ~P25;

    P1REN |= BIT7; // Pulldown P1.7
    P1OUT &= ~BIT7; // Pulldown P1.7
    P2REN |= P22; // Pulldown P2.2
    P2OUT &= ~P22; // Pulldown P2.2

    P2DIR |= P20;
    P2OUT &= ~P20;

    // ELMO
    P2DIR |= BIT1; // P2.1 is output
    P2SEL |= BIT1; // Set P2.1 to output PWM
    P2SEL &= ~BIT6; // P2.6 not used at crystal osc
    P2DIR |= BIT6; // Set P2.6 to output
    P2OUT &= ~BIT6; // Clear P2.6

}

//ADC setup function
void setup_ADC()
{
    ADC10CTL1 = INCH_6 | CONSEQ_1; // A6 + A5 + A4 + A3 + A2 + A1 + A0,
single sequence
    ADC10CTL0 = ADC10SHT_2 | MSC | ADC10ON | ADC10IE;
    while (ADC10CTL1 & BUSY);
    ADC10DTC1 = 0x07; // 8 conversions

```

```

        ADC10AE0 |= (BIT0 | BIT1 | BIT2 | BIT3 | BIT4 | BIT5 | BIT6); // ADC10
option select
}

/*
 * Triggers fresh cycle of the ADC to read the sequence of channels specified
in
 * setup_ADC(). Stores the samples in adc_sample[] in reverse order.
 */
void read_analog()
{
    ADC10CTL0 &= ~ENC;
    while (ADC10CTL1 & BUSY);
    ADC10SA = (unsigned)&adc_sample[0]; //RAM Address of ADC Data, must be
reset every conversion
    ADC10CTL0 |= (ENC | ADC10SC); //Start ADC Conversion
    while (ADC10CTL1 & BUSY); //Wait while ADC is busy
}

/*
 * Running average of the last number of samples specified by TOTAL_SAMPLES.
 * Averages for one single cell.
 */
unsigned single_cell_avg(int cell_num)
{
    unsigned average;
    unsigned cell_sum=0;
    int i;
    for(i = 0 ; i < TOTAL_SAMPLES ; i++)
    {
        cell_sum = cell_sum + adc_arr[cell_num][i];
    }
    average = cell_sum/TOTAL_SAMPLES;
    return average;
}

/*
 * takes the average of all cells excluding the cell marked by wrt
 */
unsigned panel_avg_wrt(int wrt, unsigned cell_ave_1, unsigned cell_ave_2,
                        unsigned cell_ave_3, unsigned cell_ave_4,
                        unsigned cell_ave_5, unsigned cell_ave_6)
{
    unsigned average_wrt;

    switch(wrt)
    {
        case 1: // cell 1
        {
            average_wrt = (cell_ave_2 + cell_ave_3 + cell_ave_4 + cell_ave_5
+ cell_ave_6)/5;
            break;
        }
        case 2: // cell 2
        {
            average_wrt = (cell_ave_1 + cell_ave_3 + cell_ave_4 + cell_ave_5
+ cell_ave_6)/5;

```

```

        break;
    }
    case 3: //cell 3
    {
        average_wrt = (cell_ave_1 + cell_ave_2 + cell_ave_4 + cell_ave_5
+ cell_ave_6)/5;
        break;
    }
    case 4: // cell 4
    {
        average_wrt = (cell_ave_1 + cell_ave_2 + cell_ave_3 + cell_ave_5
+ cell_ave_6)/5;
        break;
    }
    case 5: // cell 5
    {
        average_wrt = (cell_ave_1 + cell_ave_2 + cell_ave_3 + cell_ave_4
+ cell_ave_6)/5;
        break;
    }
    case 6: // cell 6
    {
        average_wrt = (cell_ave_1 + cell_ave_2 + cell_ave_3 + cell_ave_4
+ cell_ave_5)/5;
        break;
    }
    }
    return average_wrt;
}

```

```

/*
 * Checks if the running average of any single cell is less than 10% below
the
 * average of the remaining cells.
 */

```

```

unsigned shaded_cell()

```

```

{
    unsigned average_1 = single_cell_avg(0);    // cell 1
    unsigned average_2 = single_cell_avg(1);    // cell 2
    unsigned average_3 = single_cell_avg(2);    // cell 3
    unsigned average_4 = single_cell_avg(3);    // cell 4
    unsigned average_5 = single_cell_avg(4);    // cell 5
    unsigned average_6 = single_cell_avg(5);    // cell 6

    unsigned average_wrt_1 = panel_avg_wrt(1,average_1,average_2,average_3,
        average_4,average_5,average_6);
    unsigned average_wrt_2 = panel_avg_wrt(2,average_1,average_2,average_3,
        average_4,average_5,average_6);
    unsigned average_wrt_3 = panel_avg_wrt(3,average_1,average_2,average_3,
        average_4,average_5,average_6);
    unsigned average_wrt_4 = panel_avg_wrt(4,average_1,average_2,average_3,
        average_4,average_5,average_6);
    unsigned average_wrt_5 = panel_avg_wrt(5,average_1,average_2,average_3,
        average_4,average_5,average_6);
    unsigned average_wrt_6 = panel_avg_wrt(6,average_1,average_2,average_3,
        average_4,average_5,average_6);
}

```

```

        if( (average_1 < (0.9*average_wrt_1) ) || (average_2 <
(0.9*average_wrt_2) ) ||
            (average_3 < (0.9*average_wrt_3) ) || (average_4 <
(0.9*average_wrt_4) ) ||
            (average_5 < (0.9*average_wrt_5) ) || (average_6 <
(0.9*average_wrt_6) ) )
        {
            return 1;
        }
        else
        {
            return 0;
        }
    }

unsigned shaded_cell_delay()
{
    while(sample_cnt < TOTAL_SAMPLES)
    {
        read_analog();
        sample_cnt++;
    }
    sample_cnt = 0;

    unsigned is_shaded = shaded_cell();
    shadedTime1 = millisCount;
    shadedTimeDiff = shadedTime1 - shadedTime0;
    shadedTime0 = shadedTime1;
    if(is_shaded == 1)                //at least one cell is shaded
    {
        if(shadedTimeTotal < 10000)
        {
            shadedTimeTotal += shadedTimeDiff;
            return 0;
        }
        else
        {
            return 1;
        }
    }
    else                            //no cell is shaded
    {
        shadedTimeTotal = 0;
        return 0;
    }
}

// YOUSAF
void Motor_FWD()
{
    P2OUT &= ~P24;    //Turns OFF P2.4
    P2OUT |= P25;     //Turns ON P2.5
}

void Motor_REV()
{
    P2OUT |= P24;     //Turns ON P2.4

```

```

        P2OUT &= ~P25;    //Turns OFF P2.5
    }

void Motor_STOP()
{
    P2OUT &= ~P24;    //Turns OFF P2.4
    P2OUT &= ~P25;    //Turns OFF P2.5
}

void VALVE_CONTROL()
{
    P2OUT |= P20;
    int v;
    for(v=0;v<5000;v++)
    {
        __delay_cycles(16000);
    }
    P2OUT &= ~P20;
}

void MOTOR_CONTROL()
{
    P2OUT |= P23;

    Motor_FWD();

    unsigned countfwd = 0;
    while(countfwd < 65000)
    {
        if(P22 & P2IN)
        {
            countfwd++;
        }
        else
        {
            countfwd = 0;
        }
    }

    //Motor_STOP();

    /*int k1,k2;
    for(k1=0;k1<2000;k1++)
    {
        __delay_cycles(16000);
    }*/

    Motor_REV();

    unsigned countrev = 0;
    while(countrev < 65000)
    {
        if(P17 & P1IN)
        {
            countrev++;
        }
        else

```



```

        {
            countrev = 0;
        }
    }

    /*int j1,j2;
    for(j1=0;j1<2000;j1++)
    {
        __delay_cycles(16000);
    }*/

    Motor_STOP();
    P2OUT &= ~P23;

}

void Timer1PWM()
{
    TA1CCTL0 = CM_0 | CCIS_0 | SCS | CAP | OUTMOD_0;
    TA1CCTL1 = CM_0 | CCIS_0 | OUTMOD_7;
    TA1CCR0 = 320;
    TA1CTL = TASSEL_2 | ID_0 | MC_1 ;
    TA1CCR1 = 160;
}

```