University of Colorado
Boulder

Photovoltaic Power Electronics
ECEA 5718 Battery Management Laboratory

Maximum Power Point Tracking

Kai I. Tam

March 29, 2022

1. Battery current and voltage sensing

- Document complete schematics of the voltage and current sensing circuits that you implemented.

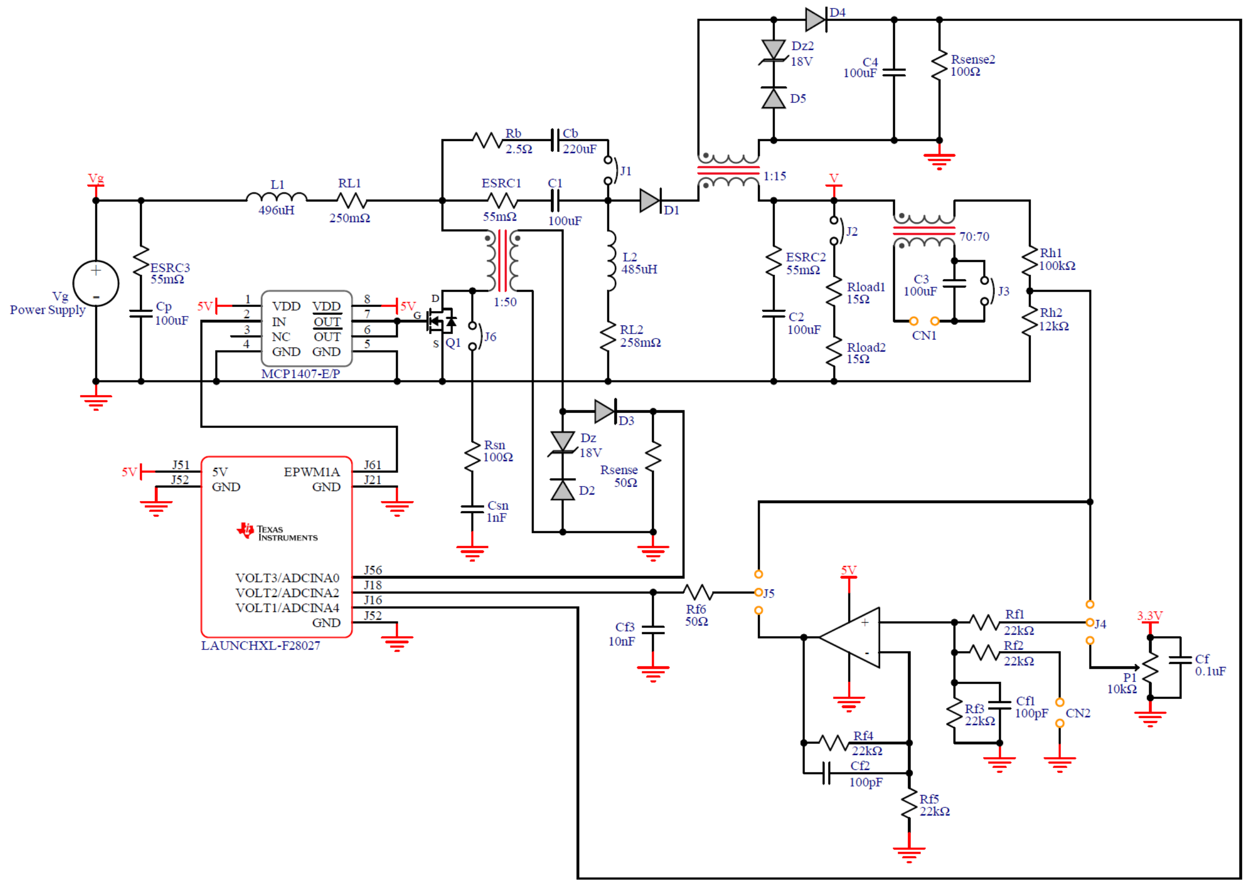
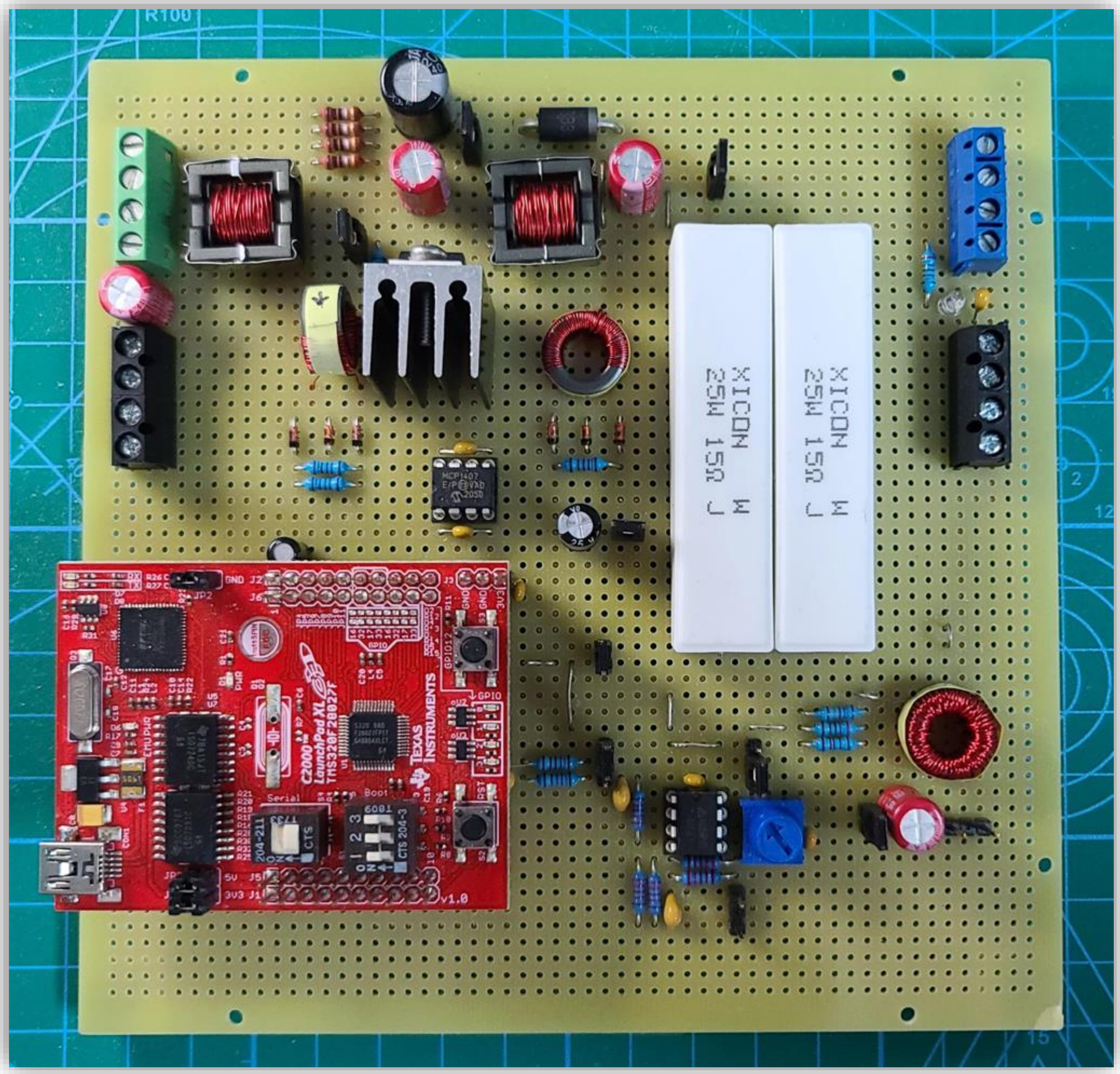


Figure 1. Complete Schematic

By utilizing the circuit that we built from last project, I just adding another current sensor, with an 1:15 effective turns ratio in series of the Diode D1. While we reuse the Voltage divider Rh1 and Rh2 as my Battery Voltage Sensing. We will bypass the conditioning op-amp with jumpers for this project.



Picture 1. Complete Circuit with Current Sensing Circuit

- Document the details of your current sense transformer design, including turns, wire gauge, and maximum flux density.

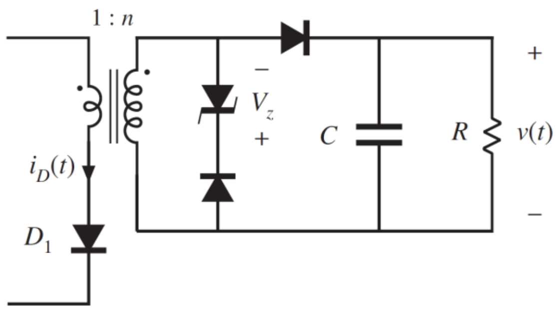


Figure 2. Current Sensing Circuit with Filtering Capacitor

D	PV Voltage	PV Current	PV Power	Battery Voltage	Battery Current	V_{batt} / V_{pv}
0.10	20.5650	0.0030	0.0617	12.5390	0.0043	0.6097
0.20	20.5790	0.0040	0.0823	12.5270	0.0057	0.6087
0.30	20.5080	0.0110	0.2256	12.5480	0.0128	0.6119
0.40	18.5490	0.4350	8.0688	12.8840	0.5652	0.6946
0.45	16.3630	0.5200	8.5088	12.6710	0.6280	0.7744
0.50	12.0410	0.3180	3.8290	12.6250	0.2592	1.0485
0.60	7.8200	0.3210	2.5102	12.4700	0.1750	1.5946
0.70	4.9700	0.3130	1.5556	12.4530	0.1060	2.5056
0.80	2.8520	0.3670	1.0467	12.4880	0.0810	4.3787
0.90	1.2710	0.4500	0.5720	12.4760	0.0329	9.8159
1.00	0.1820	0.3740	0.0681	12.4620	0.0010	68.4725

Table 1. Open Loop Battery charging data from Project 3.

Based on observation from the data of Project 3, the Peak Power would be in Duty Cycle of 0.4 < D < 0.5, with the measured current approximate as 0.63A. It can also be as high as 0.82A with 15Ω Resistive Load, so maximum resultant ADC voltage should be taken care of, so it would not exceed 3.3V. I chose a conservative value of 1A equal 3.3V as my baseline default value. The physical turns of current transformer is 1:50

$$v_{average} = \frac{I_{out}(t)}{n} \cdot R_{sense} \cdot D' \approx 3.2V$$

$$R_{sense} \approx \frac{v_{average} \cdot n}{D' \cdot I_{out}(t)} \approx \frac{3.3 \cdot 50}{0.55 \cdot 1} \approx 300\Omega$$

With the above resistor, my output voltage goes way over 3.2V. Reason could be the geometric winding of the primary turns is not exactly 1, and it is more like 1.5~2 turns. (check Picture 2 for reference) Therefore, I had to find out the effective turns ratio of my current transformer and adjust the proper Rsense value. Using a smaller resistance 100Ω, which would lower down my Vadc to an optimum level.

Condition				Output						
Vout	**Rload**	**Iout**	**D**	**Rsense**	**Vpeak**	**D'**	**Vaverage**	**Vadc**	**Turn (eff)**	**Gain (eff)**
12.92	75	0.172	0.434	100	1.28	0.520	0.666	0.640	13.46	3.86
12.76	60	0.213			1.52	0.530	0.806	0.800	13.99	3.79
12.57	45	0.279			1.92	0.535	1.027	1.040	14.55	3.68
12.3	30	0.410			2.80	0.548	1.534	1.520	14.64	3.74
11.97	20	0.599			4.00	0.553	2.212	2.240	14.96	3.70
11.85	18	0.658			4.48	0.550	2.464	2.480	14.69	3.74
11.68	15	0.779			5.20	0.553	2.876	2.960	14.97	3.69
11.14	10	1.114			7.36	0.555	4.085	4.160	15.14	3.67

Table 2. Current Sensor measurements

Measure values with Vin, D, Rsense set as constant.
Based on the actual results, the effective turns of my Current sense transformer with

$$n_{eff} \approx \frac{I_{out} R_{sense}}{V_{peak}} = \frac{0.779 \cdot 100}{5.2} = 14.98$$

$$1{:}\, n_{eff} \approx 1{:}\,15$$

Vadc is the actual voltage my ADC channel would measure and is captured with scope, where *Vaverage* is a calculated value:

$$v_{average} \approx V_{peak} \cdot D'$$

Note: We can also observe that at current above 1A, the output can be higher than 3.3V, the ADC input can be permanently damaged if directly connected to it. Even though our Set up would not be able to produce that much current (Assuming Solar Panel producing 100% output, SEPIC circuit running with 100% efficiency, and Battery at close to 0% SOC), which we will put a safe buffer in software to make sure it is current limited. As a sanity check.

The Primary wire gauge: 22AWG
The Secondary wire gauge: 30AWG

75 Material Characteristics:

Property	Unit	Symbol	Value
Initial Permeability @ B < 10 gauss		μ_i	5000
Flux Density @ Field Strength	gauss oersted	B H	4300 5
Residual Flux Density	gauss	B_r	1400
Coercive Force	oersted	H_c	0.16
Loss Factor @ Frequency	10^{-6} MHz	$\tan\delta/\mu_i$	15 0.1
Temperature Coefficient of Initial Permeability (20 -70°C)	%/°C		0.6
Curie Temperature	°C	T_c	>140
Resistivity	Ω cm	ρ	3×10^2

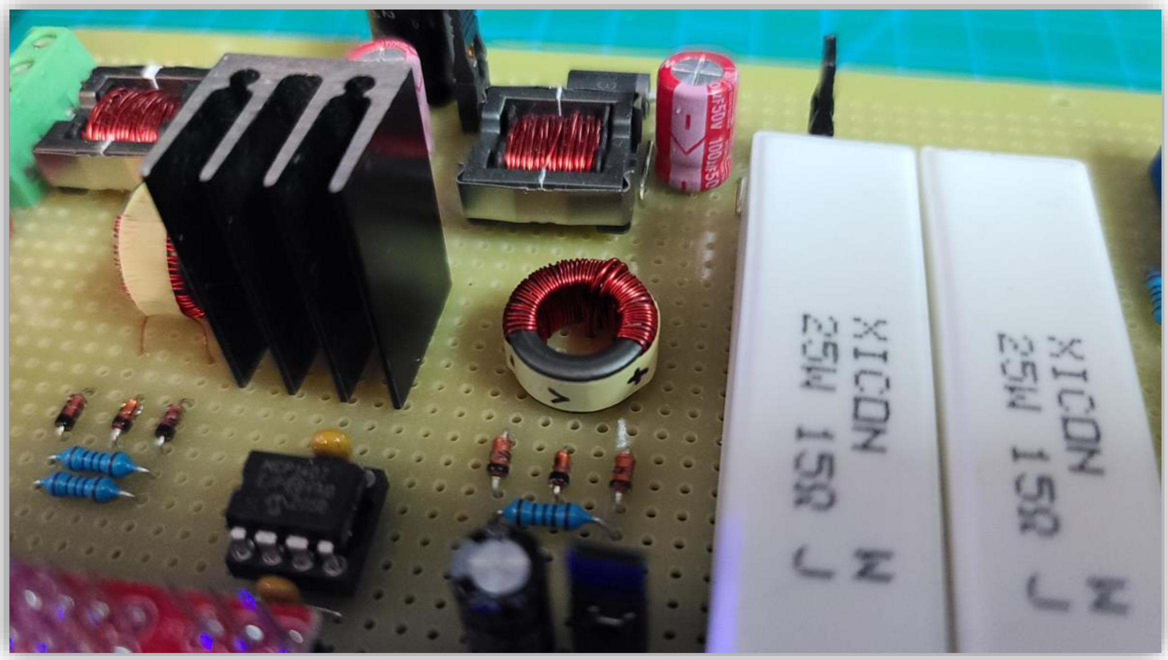
High Permeability, 75 (μ_i=5000) material

Part Number	A	B	C	Wt. (g)	$\Sigma\ell/A(cm^{-1})$	$\ell_e(cm)$	$A_e(cm^2)$	$V_e(cm^3)$	$A_L(nH)$
5975000211	9.74 Max 0.383 Max	4.56 Min 0.180 Min	3.34 Max 0.132 Max	0.83	28.60	2.07	0.072	0.15	2200 ±20%
5975000221	10.20 Max 0.401 Max	4.10 Min 0.162 Min	3.80 Max 0.149 Max	0.83	28.60	2.07	0.072	0.15	2200 +20%, -25%
5975000301	12.70 ±0.25 0.500	7.15 ±0.20 0.281	4.90 -0.25 0.188	2.00	22.90	2.95	0.129	0.38	2725 ±20%
5975000321	13.45 Max 0.529 Max	6.45 Min 0.254 Min	5.40 Max 0.212 Max	2.00	22.90	2.95	0.129	0.38	2725 +20%, -25%
5975001101	12.70 ±0.25 0.500	7.90 ±0.20 0.312	6.35 ±0.25 0.250	2.40	20.80	3.12	0.15	0.47	3000 ±20%

$$I_{peak} = 1A\ (conservative\ number)$$
$$n \approx 1$$
$$l_m = 3.12cm = 0.0312m$$
$$\mu_o = 5000$$

The maximum flux density:

$$B_{max} = \frac{I_{peak} \cdot n \cdot \mu_o \cdot \mu_r}{l_m} = \frac{1 \cdot 1 \cdot 5000 \cdot 4\pi \cdot 10^{-7}}{0.0312} \cong 0.201384T$$



Picture 2. Output Current (Diode D1 Current) Sensor Circuit

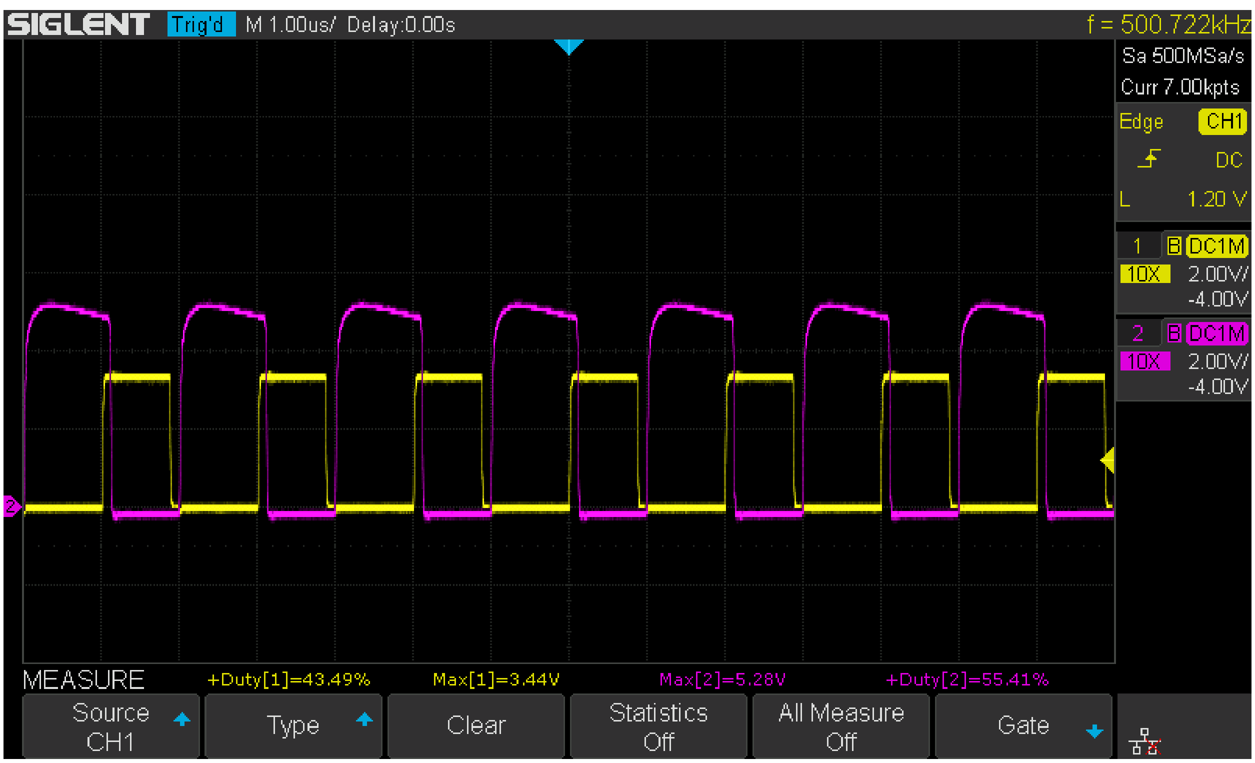


Figure 3. Ch1(PWM), Ch2(Sensor output without Capacitor Filtering)

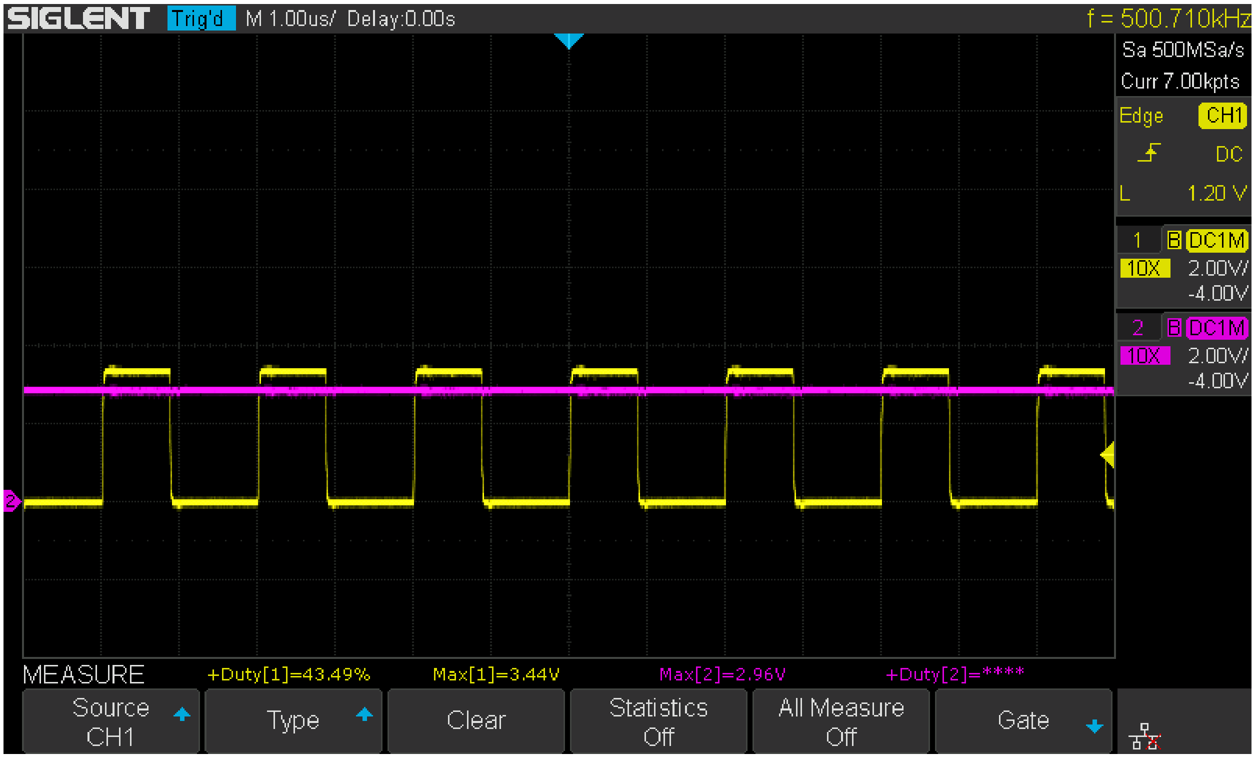


Figure 4. Ch1(PWM), Ch2(Sensor output without Capacitor Filtering)

Without the filtering capacitor, the peak can go over 3.3V, but we can further observe the Diode D1 current waveform. Which happened at D', we can also observe the negative current slope ΔI.

With the Filtering Capacitor 100μF, the output is in smooth DC.

- Include the plot of measured current sensor characteristic and the gain as defined in part 1 Above.

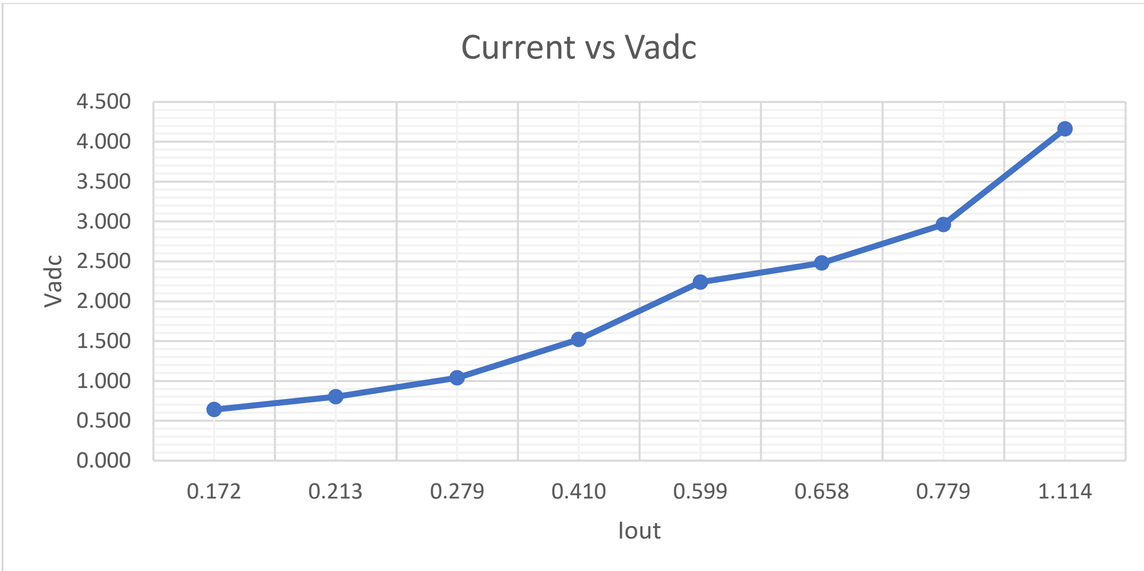


Figure 5. Current vs Vadc

Condition				Output						
Vout	**Rload**	**Iout**	**Duty**	**Rsense**	**VPeak**	**Duty**	**Average**	**Vadc**	**Turn (eff)**	**Gain (eff)**
12.92	75	0.172	0.434	100	1.28	0.520	0.666	0.640	13.46	3.86
12.76	60	0.213			1.52	0.530	0.806	0.800	13.99	3.79
12.57	45	0.279			1.92	0.535	1.027	1.040	14.55	3.68
12.3	30	0.410			2.80	0.548	1.534	1.520	14.64	3.74
11.97	20	0.599			4.00	0.553	2.212	2.240	14.96	3.70
11.85	18	0.658			4.48	0.550	2.464	2.480	14.69	3.74
11.68	15	0.779			5.20	0.553	2.876	2.960	14.97	3.69
11.14	10	1.114			7.36	0.555	4.085	4.160	15.14	3.67

Table 2. Current Sensor measurements

Based on Table 2, the averaged effective gain is:

$$Gain_{eff} = \frac{V_{adc}}{I_{out}} \approx 3.73$$

We can observe the plot is positive proportional. Since we are not measuring exact current value, exact linearity is not necessarily needed for MPPT, all we need is the positive proportional relationship.

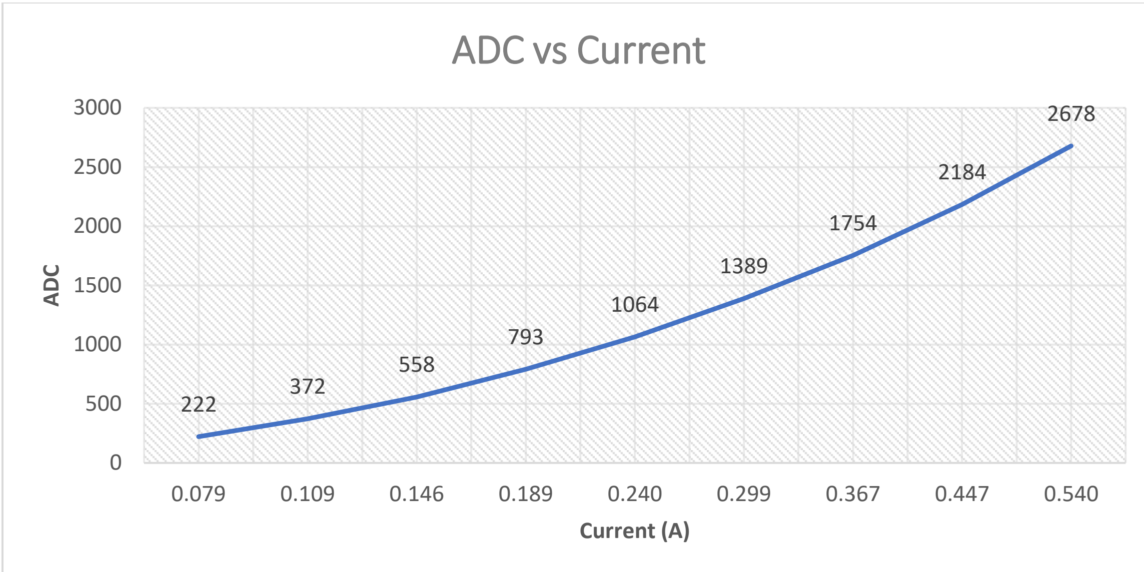


Figure 6. Current vs Vadc

Condition				Voltage			Current		
Input V	d	Real d	Load	Output V	Vadc	V gain	Output A	Aadc	A gain
17	5	5.78	30	1.549	250	161	0.052	143	2770
	10	10.74		2.360	350	148	0.079	222	2822
	15	15.71		3.277	475	145	0.109	372	3406
	20	20.68		4.375	617	141	0.146	558	3826
	25	25.65		5.679	789	139	0.189	793	4189
	30	30.62		7.201	982	136	0.240	1064	4433
	35	35.59		8.965	1223	136	0.299	1389	4648
	40	40.56		11.017	1498	136	0.367	1754	4776
	45	45.53		13.410	1805	135	0.447	2184	4886
	50	50.50		16.204	2165	134	0.540	2678	4958

Table 3. Voltage-ADC gain, Current-ADC gain

- Document the interrupt service routine code that measures battery current and voltage

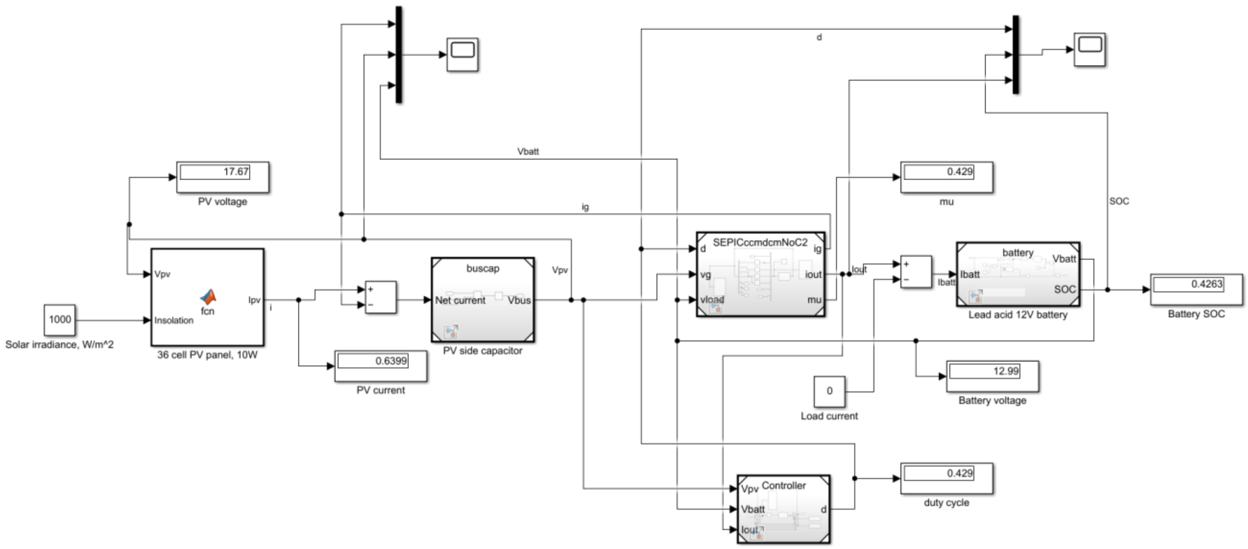
```
adc_isr(void)
{
//  GPIO19 Toggle
        GPIO_setHigh(myGpio, GPIO_Number_19);                   // Toggle GPIO high to find out the Sampling Time.

//  Reading
        Voltage1 = ADC_readResult(myAdc, ADC_ResultNumber_1);   // Reading Battery Charging Current
        Voltage2 = ADC_readResult(myAdc, ADC_ResultNumber_2);   // Reading Battery Voltage

//  Set PWM duty cycle (below are setting dummy value for circuit measurement.)
        PWM_setCmpA(myPwm, 51);                                 // Set compare A value
        PWM_setCmpAHr(myPwm, (unsigned int)(1 << 8));           //

//  GPIO19 Toggle
        GPIO_setLow(myGpio, GPIO_Number_19);                    // Toggle GPIO low to find out the Sampling Time.

//  Interrupt Reset
        ADC_clearIntFlag(myAdc, ADC_IntNumber_1);               //  Clear ADCINT1 flag reinitialize for next SOC
        PIE_clearInt(myPie, PIE_GroupNumber_10);                //  Acknowledge interrupt to PIE

    return;
}
```

This preliminary code can be used to verify the sampling time and current sensor measurements.

With Power Supply set to constant 17V and constant 30Ω Load, adjust Duty Cycle (d) with function *PWM_setCmpA(myPwm, ??)*, where ?? out of 120 (our period of 500kHz switching frequency)

2. MATLAB/Simulink model

- Document the code that implements the maximum power point tracking routine of your Simulink model




```
function Ipv = fcn(Vpv,Insolation)
%#codegen
ncells = 36;          % number of PV cells, checked
ki = 7.09e-4;         % gain from insolation to current, checked.
%IF = 5e-10;          % diode saturation current
IF = 2.14e-10;        % diode saturation current
%lambda = 36;         % q/kT
lambda = 38.46;       % q/kT, 1/0.026
Rshunt = 500;         % shunt resistance, checked
Ipv = ki*Insolation - IF*(exp(lambda*Vpv/ncells) - 1) - Vpv/Rshunt;
```

Matlab 1. Solar Panel model parameters.

```
function y = fcn(u)
%#eml
% SEPIC state-space model with CCM-DCM averaged switch model
L1 = 496e-6;          % inductance L1
L2 = 485e-6;          % inductance L2
RL1 = 250e-3;         % L1 winding resistance
RL2 = 258e-3;         % L2 winding resistance
C1 = 100e-6;          % capacitance C1
Cd = 220e-6;          % damping capacitor Cd
Rd = 2.5;             % damping resistor Rd
Ts = 2e-6;            % switching period
…
…
```

Matlab 2. SEPIC model parameters.

```matlab
function y = fcn(u)
%% PV system controller
%
%  Enter your battery charge controller algorithm and parameters in this
%  function
%
%% input signals (sampled)
%
Vpv = u(1);          % sensed PV voltage (might or might not be used in MPPT algorithm)
Vbatt = u(2);        % sensed battery voltage (needed for charge taper mode)
Iout = u(3);         % sensed SEPIC output current (might be used in MPPT algorithm)
lastduty = u(4);     % MPPT last duty cycle
lastdir = u(5);      % MPPT last direction: +1 or -1 (+1 means duty cycle was increased)
lastI = u(6);        % MPPT last converter output current value
wait = u(7);         % variable used to count settling periods after change in duty cycle
lastmode = u(8);
%
%% Algorithm parameters, can be adjusted to tune control performance
%
% Define your algorithm parameters here
    settlePeriods = 10;      % number of sample periods to wait for system to settle
    step = 0.001;           % MPPT algorithm duty cycle step size
    Dmax = 0.9;             % limit max duty cycle of SEPIC
%
%% Control algorithm
%
% Enter your control algorithm here
mode = lastmode;
    if wait <=0
        wait = settlePeriods;
            if Iout < lastI
                direction = -lastdir;
            else
                direction = lastdir;
            end
        d = min(Dmax, max(0,lastduty+direction*step));
        lastI = Iout;
    else
        wait = wait-1;                  % decrement wait (count variable)
        d = lastduty;                   % pass duty cycle to next sampling period
        direction = lastdir;            % pass direction to next sampling period
    end
%
% vector output of this function
y = [d direction lastI wait mode];
```

Matlab 3. PV System Controller

- Include your simulation scope waveforms of duty cycle and SEPIC output current, showing the system finding and maintaining the maximum power point, as described in part 2 above

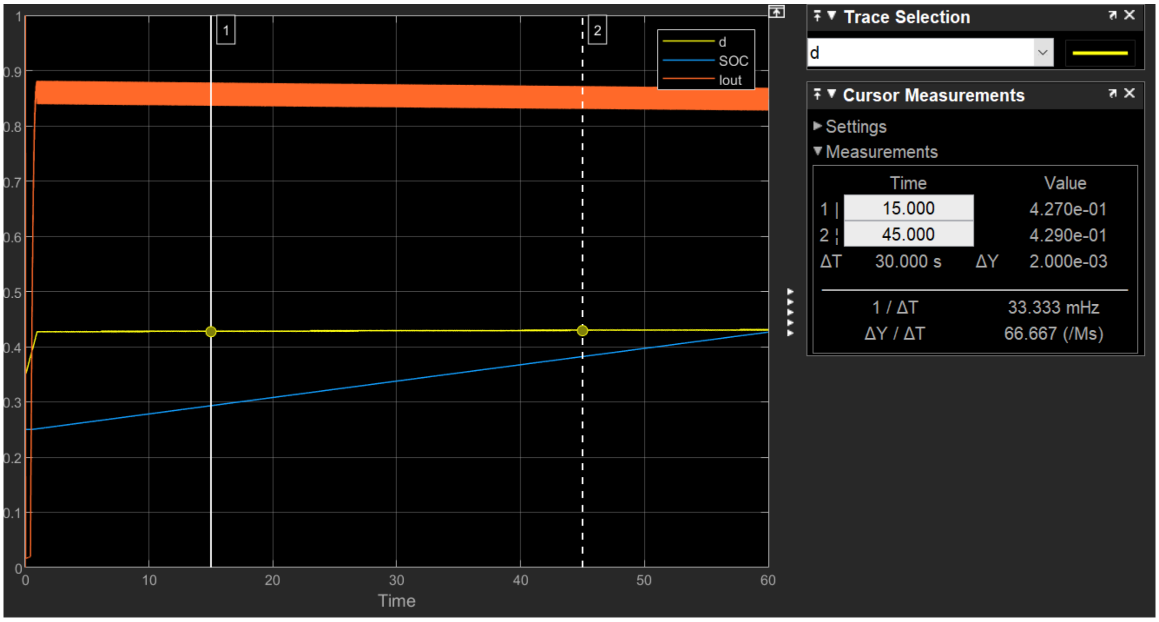


Figure 7. d SOC Iout (1 minute simulation time)

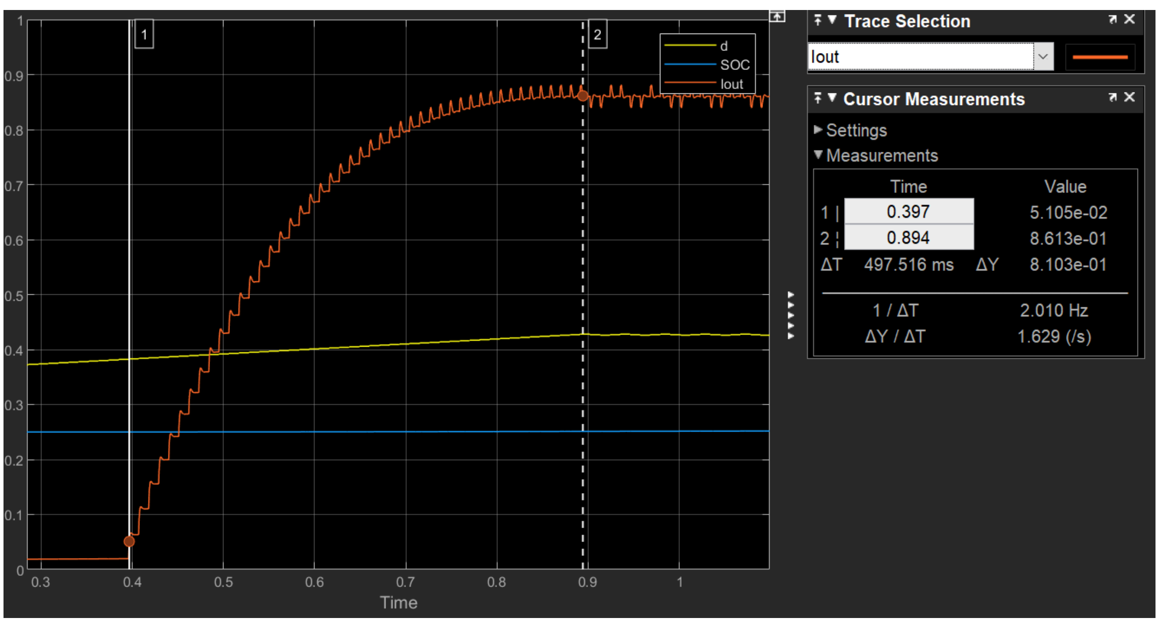


Figure 8. d SOC Iout (Time it takes to reach MPP)

We can observe the SOC is gradually increasing with Duty Cycle (d) almost at constant 0.43. Iout is fluctuating and finding its maximum peak constantly.

- Document the values of (i) the sampling period, (ii) the number of sampling periods employed for settling, (iii) the time required to reach the maximum power point.

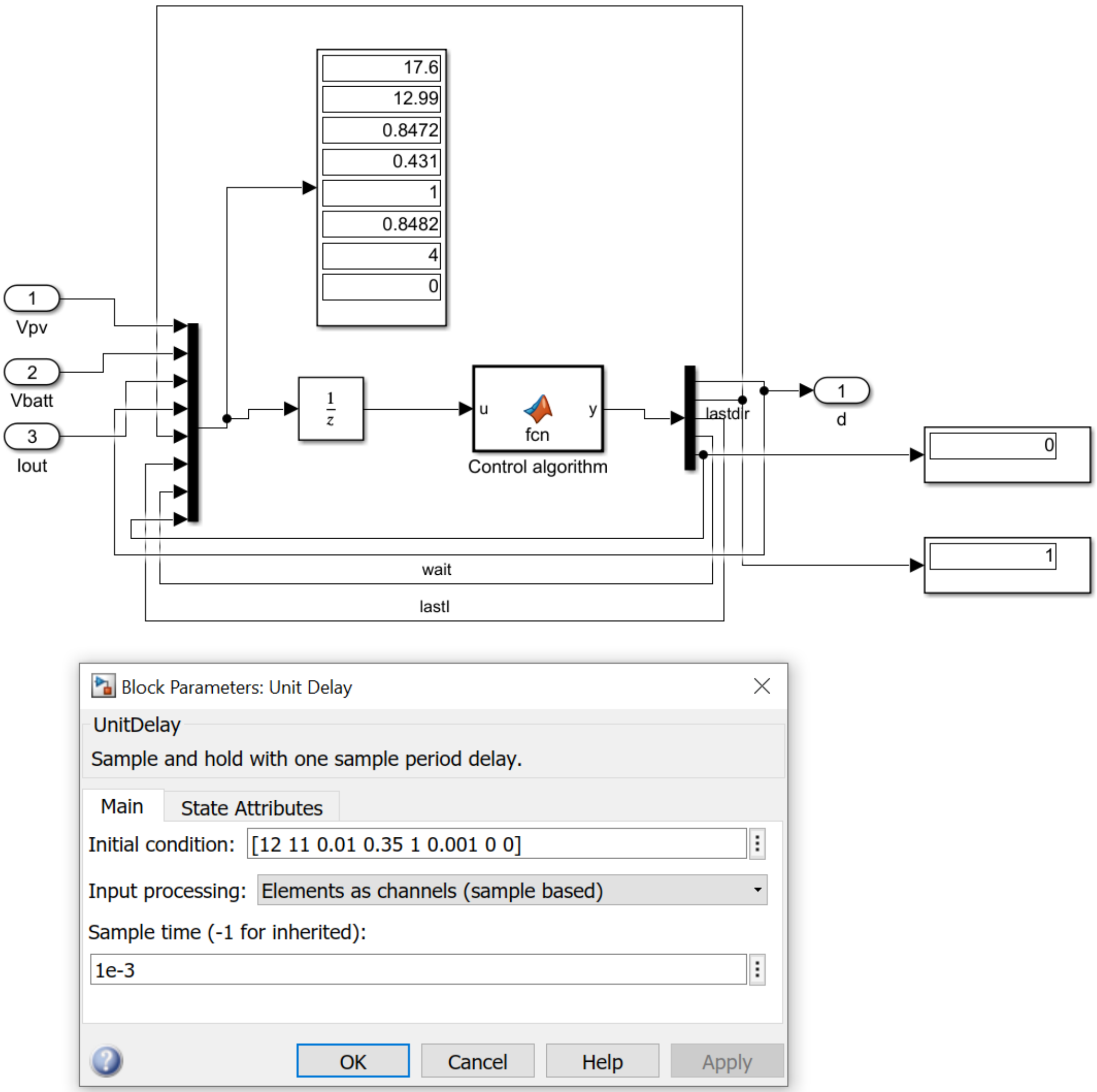


Figure 9. Controller sample readings.

Sample time:	1mS
Number of sampling periods employed for settling:	10
Time required to reach the maximum power point.	497mS

3. Implementation of MPPT

- Document the C code that implements your hardware maximum power point tracking algorithm

```
//
// Globals
//
uint16_t                LoopCount;
uint16_t                ConversionCount;
uint16_t    volatile    Voltage2;
uint16_t    volatile    DutyCycle;
uint16_t    volatile    DutyCycle_p;

uint32_t    volatile    counter             =   0;          // For extending ADC sampling time.
uint16_t    volatile    direction           =   1;          // 1 increasing (default), 0 decreasing

int64_t     volatile    last_power          =   0;          // not used in MPPT
int64_t     volatile    power;                              // not used in MPPT
int64_t     volatile    last_control        =   10;         // Start up duty cycle with 10.
int64_t     volatile    control             =   10;         // Start up duty cycle with 10.

int32_t     volatile    last_voltage        =   0;          // not used in MPPT
int32_t     volatile    voltage;                            // not used in MPPT
int32_t     volatile    last_current        =   0;          // last current value for comparison
int32_t     volatile    current;                            // Charge / output Current ADC value

uint16_t    volatile    increment           =   1;          // increment of Duty Cycle (1/120)
```

...

```
//
//  adc_isr -
//
#pragma CODE_SECTION(adc_isr, "ramfuncs");
__interrupt void
adc_isr(void)
{
        counter = counter + 1;                                              // counting up

        if (counter >= 14000)                                               // change settling time here (14000 = 40mS til next ADC
reading)
            {
            // GPIO19 Toggle
                    GPIO_setHigh(myGpio, GPIO_Number_19);                   // Toggle GPIO high to find out the Sampling Time.

            // Reading
                    current = ADC_readResult(myAdc, ADC_ResultNumber_1);    // Reading Battery Charging Current
                    voltage = ADC_readResult(myAdc, ADC_ResultNumber_2);    // Reading Battery Voltage

                    if ( current < last_current )
                        {
                            direction = 0;                                  // for monitoring purpose only
                            control = last_control - increment;
                        }
                    else
                        {
                            direction = 1;                                  // for monitoring purpose only
                            control = last_control + increment;
                        }

            // Reset Duty Cycle
                    if ( (control >= 70) || ((control <= 1)) )              // Reset Duty Cycle when out of bound
                        {
                            control = 10;
                        }

            // Set PWM duty cycle
                    PWM_setCmpA(myPwm, control);                            // Set compare A value
                    PWM_setCmpAHr(myPwm, (unsigned int)(1 << 8));           //

            // Push state
                    last_control    = control;                              // Pushing States
                    last_current    = current;                              // Pushing States

            // GPIO19 Toggle
                    GPIO_setLow(myGpio, GPIO_Number_19);                    // Toggle GPIO low to find out the Sampling Time.

            // Reseting counter
                    counter = 0;                                            // reset counter
            }

    // Interrupt Reset
        ADC_clearIntFlag(myAdc, ADC_IntNumber_1);                           // Clear ADCINT1 flag reinitialize for next SOC
        PIE_clearInt(myPie, PIE_GroupNumber_10);                            // Acknowledge interrupt to PIE

        return;
}
```

- Document your oscilloscope waveforms of the PV panel voltage and current sensor output voltage, showing the perturb-and-observe or other MPPT algorithm behavior around the maximum power point.

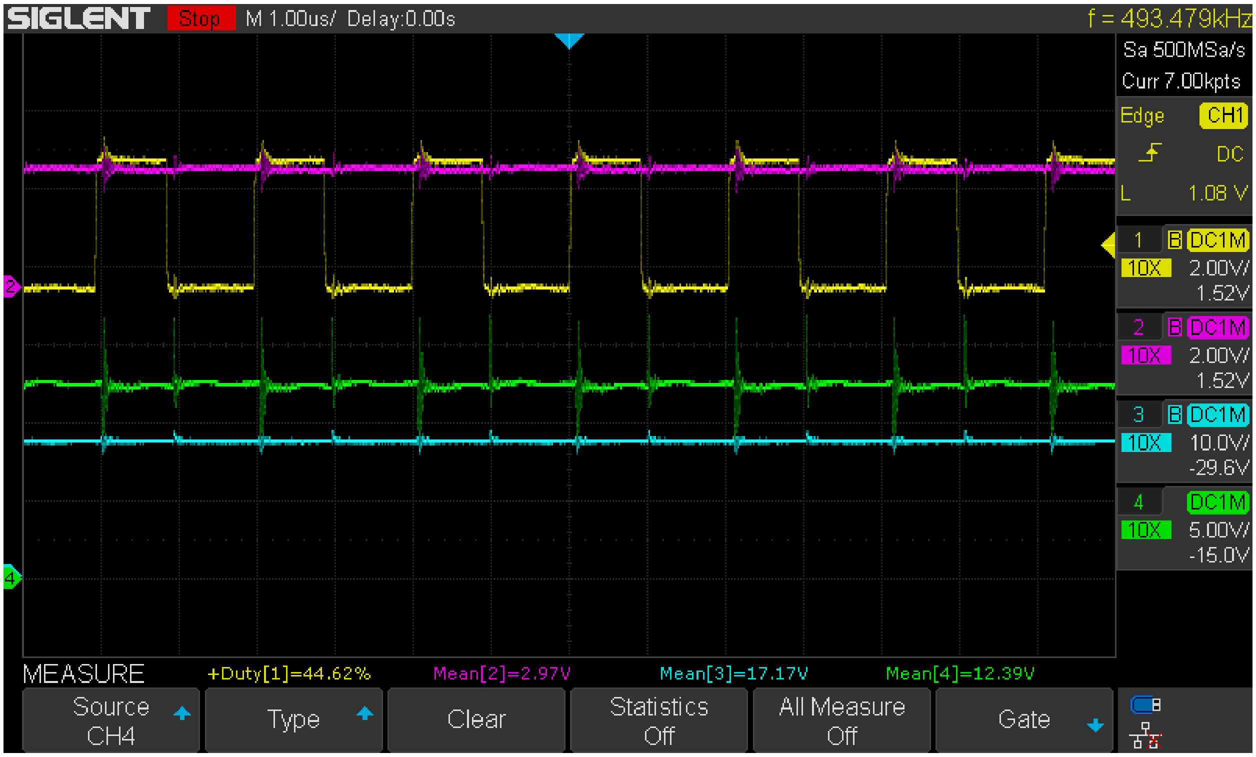


Figure 10. MPPT scope readings.

Channel 1: PWM from Launchpad.
Channel 2: Output of Current Sensor.
Channel 3: Solar Panel Voltage.
Channel 4: Battery Voltage.

03/29/22	Solar Panel		LaunchPad IO			Battery	
Time	**Vg (V)**	**Ig (A)**	**d**	**Vadc (V)**	**Value adc**	**Vbatt (V)**	**Icharge (A)**
12:15 pm	17.5	0.67	0.44	2.71	3700	12.45	0.8

The reading was captured at 03/29/2022 12:15pm Middle Village NY, Mostly Sunny at the time of measurement. We can observe that the duty Cycle stays around 0.43-0.44 which is very close to our duty cycle (0.45) of maximum delivery output measured in Project 3 (Table 1).
The current sensor reading 2.96V, which is around 0.8A based on our average conversion ratio (3.73 : 1), Battery voltage is 12.39V, the overall output power is 9.912W.
Channel 3 would be fluctuating up and down in small value around 17.17 to 17.5V at the time of operation (perturb and observe). The input current measured is around 0.67A, The Solar Panel is outputting approximate 11.6W.

- Compare the average charging battery current that your MPPT system achieves, vs. the battery current achieved with direct energy transfer.

V_{pv} (V)	I_{pv} (A)	P_{pv} (W)
0.0000	0.7089	0.0000
5.3746	0.7084	3.8071
10.3220	0.6884	7.1051
16.4090	0.6418	10.5305
17.3910	0.5813	10.1096
18.9060	0.3174	6.0011
20.5260	0.0000	0.0000

Table 4. PV Panel experimental data from Project 1.

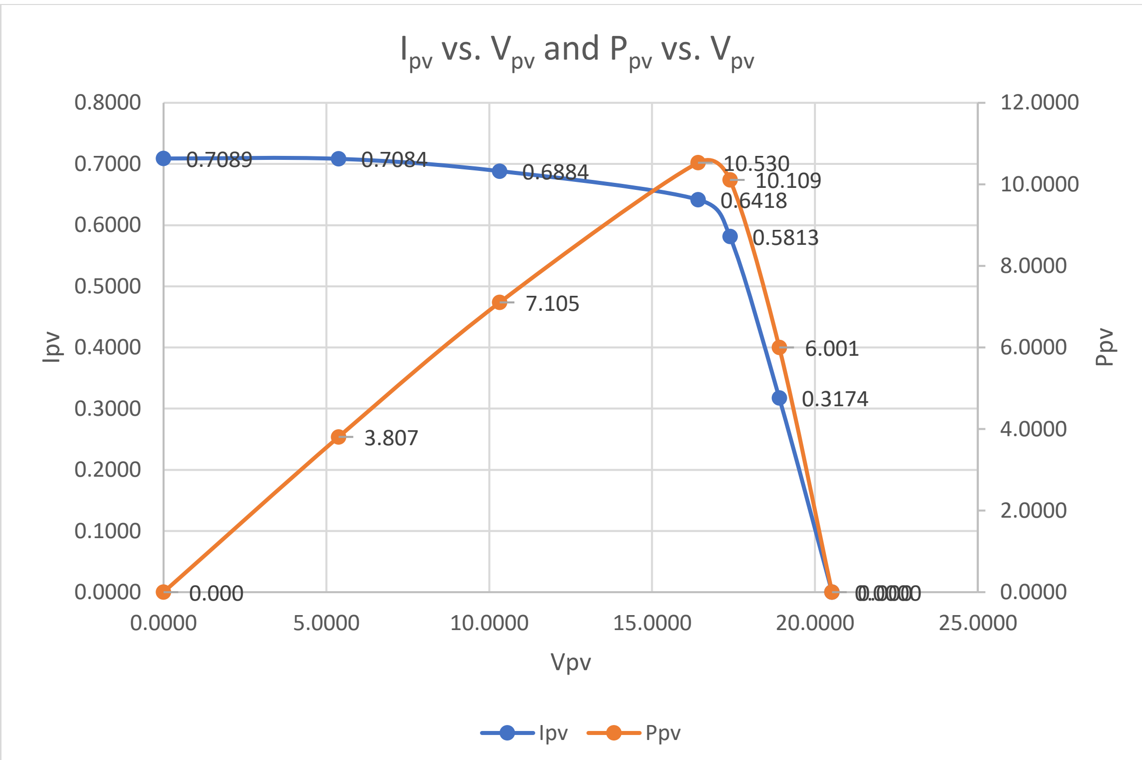


Figure 11. PV Panel experimental data from Project 1.

In Project 1, the battery current achieved with direct energy transfer happened at 16.4V and 0.6418A, which is around the peak of the power plot. The final SEPIC circuit can output as high as 0.8A which is higher than the direct energy transfer approach. While also maintaining a proper Battery Voltage.

Picture 3. Testing Set up 03/29/2022 12:15pm Middel Village, NY (mostly Sunny)

Measurement Equipment:

- FLUKE 289 TRUE RMS MULTIMETER (Calibrated by TRANSCAT on 06/15/2021)
- SIGLENT SDS 1104X-E DIGITAL STORAGE OSCILLOSCOPE
- EXTECH 1430 TRUE RMS MULTIMETER