Background

The goal of Experiment 3 Part 1 is to design, construct, test and demonstrate the buck dc-dc converter power stage discussed in lecture, under open-loop conditions.

Laboratory Procedure

1. **Inductor design and construction**
   Your laboratory kit contains several ferrite cores. Datasheets describing the important parameters of this core shape and core material are available on the course website in the Experiment 3-1 folder. Copper magnet wire of various gauges, as well as shim stock for gapping the core, are available in the lab. Construct the inductor you designed in your pre-lab assignment: select the wire gauge, wind the required number of turns on the bobbin, insert the proper air gaps, and assemble the core. Check the inductance value L on the RLC meter, and iterate if necessary.

2. **Buck power stage construction**
   Mount the power MOSFET and Schottky diode on the heat-sinks from your lab kit. Use the insulators in your kit to insulate the MOSFET and diode cases from the heatsinks; a very thin layer of thermal paste (provided by TAs) is needed on both sides of the insulator to ensure good thermal conduction. A drill will be provided, for drilling holes in your kit "perf board" to mount standoffs. Construct the power stage as shown in Fig. 1 on the perf board. Use #18 AWG wire to make interconnections in the power stage. For the MOSFET, inductor, and diode, leave loops of wire long enough to insert a clip-on AC current probe to measure i_L, i_T, i_C, and i_D. Otherwise, keep the wiring short for connections having pulsating currents. Use twisted pairs to make the signal and return connections between boards.

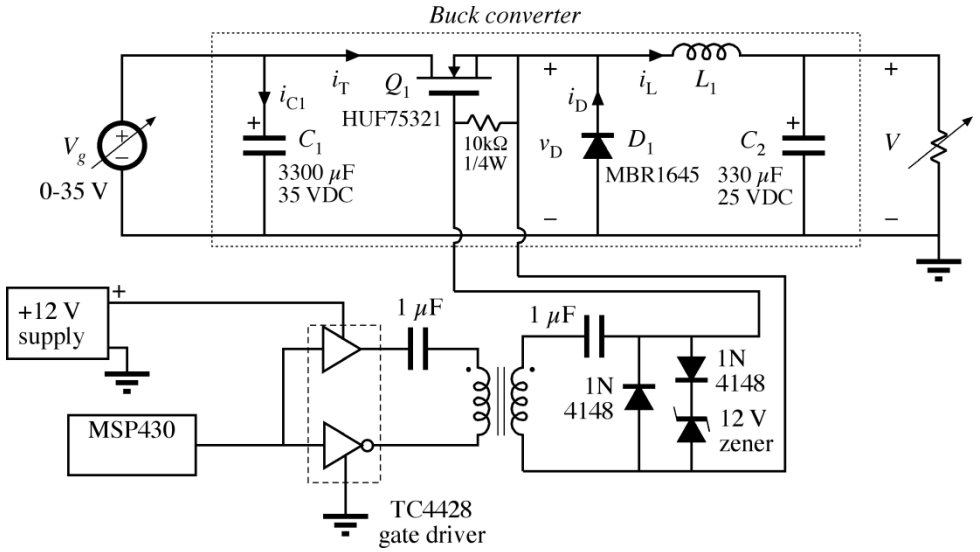


Fig. 1

3. Buck power stage testing

Connect a logic output of your MSP430 through the gate driver IC as shown, to drive the gate of the power MOSFET. The schematic of the TC4428 is available on the course website in the Experiment 3-1 folder; be sure to include a capacitor to bypass the power supply of the TC4428. Connect a load resistor, capable of consuming the power produced by one PV panel, to the converter output. Program the MSP430 to produce a suitable duty cycle signal. Apply power to the gate driver IC, and verify that the correct gate drive signal is present at the MOSFET.

With all of the power supplies turned off, connect the power stage to the laboratory bench 0-35 VDC power supply (to supply V_g). Use a voltage probe to measure $v_D(t)$, and a current probe to measure the transistor current $i_T(t)$. Use multimeters to measure the load resistor voltage and current. Set the load resistor to its maximum value, set V_g to zero, then turn on all power supplies. Increase V_g to a few volts, and verify that the $v_D(t)$ and $i_T(t)$ waveforms are correct. Be sure not to exceed the 25 V rating of output capacitor C_2. Slowly increase V_g to the rated full-power voltage of the PV panels, adjust the duty cycle to obtain an output of 13 V, and slowly decrease the load resistance until the output power is 85 W. Record the dc voltage and current of the source V_g and the load, and calculate the efficiency. If these readings make sense, proceed.

Measure the following waveforms, and record for your report:

- Diode voltage $v_D(t)$
- Transistor current $i_T(t)$
- Diode current $i_D(t)$
- Inductor current $v_D(t)$
- Capacitor C_1 current $i_{C1}(t)$
- Capacitor C_1 voltage $v_{C1}(t)$

Label the waveform names and scales. Label salient features.

Measure the inductor current ripple Δi_L, and compare with the value you designed for in your pre-lab assignment. Include these values in your report.

Measure the capacitor voltage ripple Δv_{C1}, and note the waveform of $v_{C1}(t)$. A practical capacitor model includes an *equivalent series resistance* (ESR) in series with an ideal capacitor. In many capacitors, including the aluminum electrolytic capacitors used in this experiment, the ESR induces the major portion of the ac voltage ripple. Based on your measured capacitor voltage and current waveforms, estimate the value of the ESR. Also estimate the power loss induced in the ESR, $P_C = I_{rms}^2 \, ESR$. This power loss limits the maximum ac current that can be handled by the capacitor. Include in your report: (*i*) your estimate of the ESR, (*ii*) your estimate of the power loss P_C, and (*iii*) the datasheet rms current rating for this capacitor.

4. Load test

Adjust or change the load resistor, to obtain a converter output power of approximately 15 W, with the duty cycle adjusted as necessary to obtain an output voltage of 13 V. Record the input and output voltages, the output current, and the duty cycle. Measure and record the

waveforms of $i_T(t)$ and $v_D(t)$; label the waveform names, scales, and salient features. Was it necessary to significantly change the duty cycle in this part? Why or why not? Repeat, for an output power of approximately 1 W.

5. Open loop behavior in PV system

If there is insufficient sunshine to operate the PV panels outside, then proceed to the Experiment 3 Part 2 procedure, and come back to this part next week.

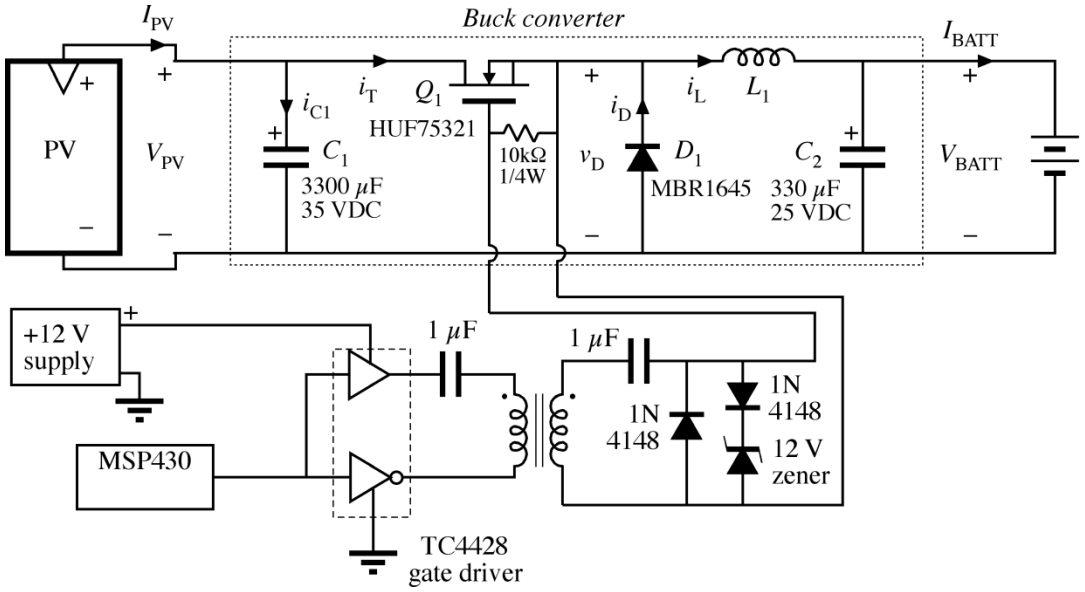


Fig. 2

Modify your programming of the MSP430 to produce a signal that is proportional to a voltage applied to an A/D input of the MSP430. See Appendix A for sample code that operates the ADC10 peripheral of the MSP430F5172. Connect a potentiometer to provide a 0–1.5 V signal at this input (don't exceed the MSP430 power supply voltage!), and verify that your MSP430 produces a signal with controllable duty cycle at its PWM output. Pay special attention to capacitor bypassing (see Appendix A): the ADC input pin will require capacitor bypassing, and longer sampling windows will also help reduce the noise.

Inside the lab, connect the converter on the PV cart, between the PV panel and the battery, as illustrated in Fig. 2. Note that the cart includes two isolated 12 VDC power supplies; one of these should be used to power your control circuitry. Connect meters to measure the PV voltage and current, as well as the battery current (you will need ammeters with 10 A or 20 A scales). While still inside the lab (with the PV panel not illuminated), turn on the power and use the oscilloscope to verify that the gate driver operates correctly. Turn off the power and take the cart (with meters and oscilloscope) outside.

Load the battery with the inverter, and use the light bulb (as well as the meters and oscilloscope) to load the inverter so that you don't overcharge the battery in this step. Point the PV panel at the sun, and power up your converter. Vary the duty cycle D in enough steps to obtain a reasonably smooth plot, from the minimum D to the maximum D that your drive

circuit can provide. Record all meters (including the battery voltage meter built into the cart) and the duty cycle.

6. Simulation and interpretation of data

In your report, you should plot your step 5 measured data of (*i*) PV voltage vs. *D*, (*ii*) battery current vs. *D*, (*iii*) PV power vs. *D*, and (*iv*) V_{batt}/V_{pv} vs. *D*. Explain theoretically the salient features of your plots.

In the lab, perform a dc simulation of the power stage of the system of Fig. 2 using SPICE (or simulink). Use your model developed in Experiment 1 for the PV panel, and use the SPICE averaged switch model (or your own simulink equivalent) described in lecture in place of the MOSFET and diode. Choose a suitable battery model. Sweep the duty cycle from 0 to 1, and make SPICE generate simulated versions of plots (*i*) to (*iv*) listed in the previous paragraph.

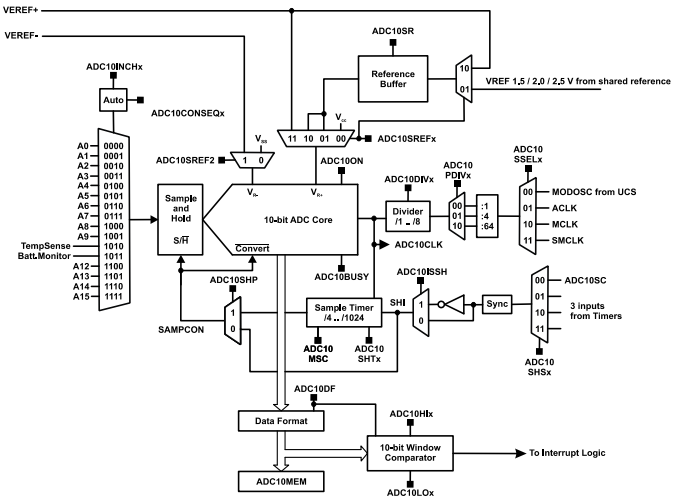
How do your measured and simulated plots compare with the theoretical model of the buck converter? If you choose the optimum duty cycle, how much power can you obtain? How does this compare with the direct energy transfer approach used to charge the battery in Experiment 1?

Appendix A – Analog to Digital (A/D) Converter of the MSP 430

ADC10: The 10-Bit A/D Converter
of the MSP430

Key features:

- Multiplexed inputs
- Sample and hold circuit
- Successive approximation register, driven by selectable clock
- Selectable reference sources
- Buffered output memory
- 10 bit or 8 bit conversion



Successive Approximations

- After the input signal has been sampled, the 10-bit SAR requires 11 clock cycles to generate an output
- Compare analog input with references
- The MSP430 uses a switched capacitor scheme to perform the comparisons
- See *MSP430x5xx Family User's Guide*, Ch. 27

Reference: John H. Davies, *MSP430 Microcontroller Basics,* Elsevier, 2008, ISBN 987-7506-8276-3.

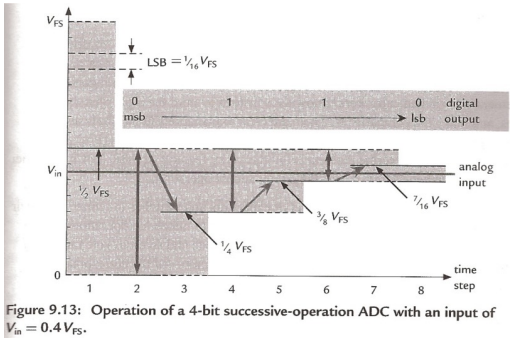


Figure 9.13: Operation of a 4-bit successive-operation ADC with an input of $V_{in} = 0.4\,V_{FS}$.

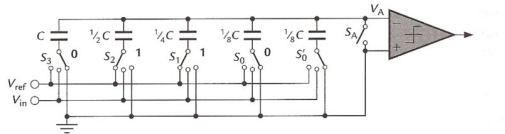
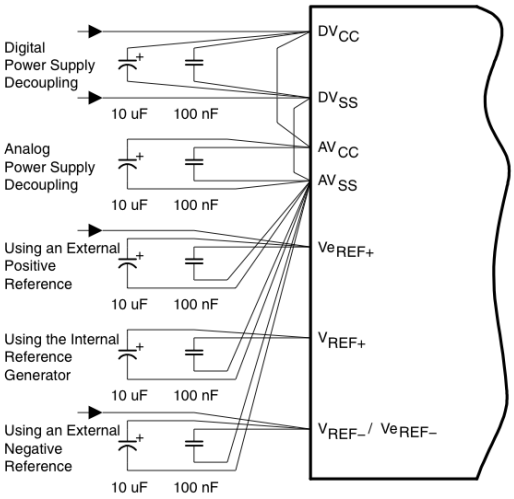


Figure 9.15: The SAR ADC with the switches in the final positions for an input of $V_{in} = 0.4\,V_{FS}$ and a binary output of 0110.

Capacitor bypassing is required

What the User's Guide recommends:

Also need capacitance (an RC low-pass filter) at each analog input pin



Setting up the A/D Converter ADC10

```
// Configure ADC10
ADC10CTL0 = ADC10SHT_2 + ADC10ON;          // sample time of 16 clocks, turn on
                                           // use internal ADC 5 MHz clock
ADC10CTL1 = ADC10SHP + ADC10CONSEQ_0; // software trigger to start a sample
                                           // single channel conversion
ADC10CTL2 = ADC10RES;                      // use full 10 bit resolution
ADC10MCTL0 = ADC10SREF_1+ADC10INCH_5; // ADC10 ref: use VREF and AVSS
                                           // input channel A5 (pin 10)
// Configure internal reference VREF
while(REFCTL0 & REFGENBUSY);               // if ref gen is busy, wait
REFCTL0 |= REFVSEL_0 + REFON;              // select VREF = 1.5 V, turn on
_delay_cycles(75);                         // delay for VREF to settle
```

The above code sets up the 10-bit ADC with A5 as its only input, with 1.5 V giving a reading of $2^{10} - 1$, and 0 V giving a reading of 0. Each reading will employ a sampling window of 16 ADC clocks = 3.2 µsec.

Sampling the ADC input

```
ADC10CTL0 |= ADC10ENC + ADC10SC;        // sampling and conversion start
while(ADC10CTL1 & ADC10BUSY);           // wait for completion
X = ADC10MEM0;                          // ADC10MEM0 contains result
```

The above code is simple and a good start. See CCS5 code examples for use of interrupts that do not require the processor to wait during the conversion time.