Today's plan:

- Announcements: status report, midterm results
- op-amp wrap-up
- Comparators
- Measuring capacitance
- Powering your project

Announcements: Status Report

- I would like a short written status report from everyone turned in at start of the third project session:

  Tue March 7

  Thu March 9

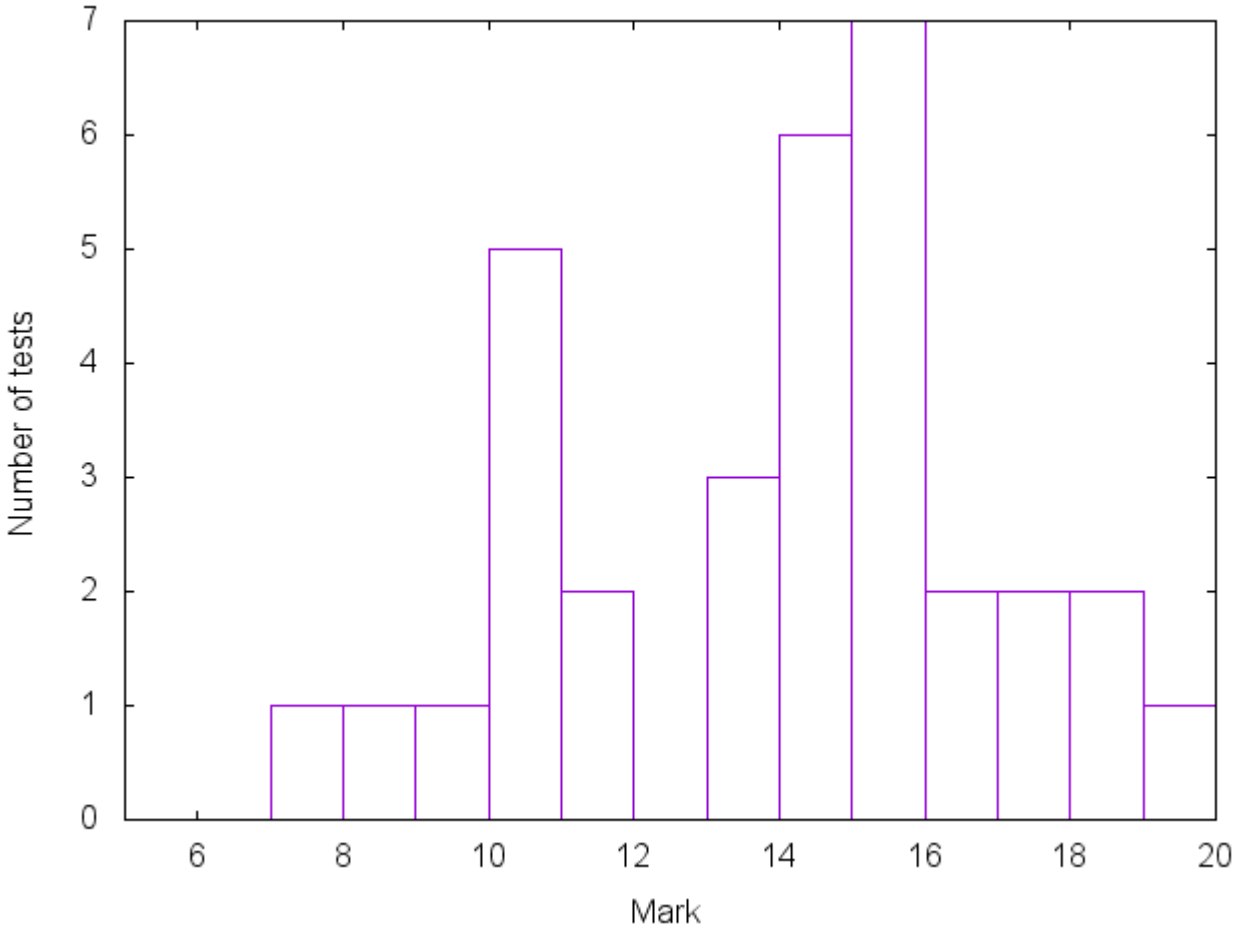
  Monday March 13

- The report should discuss your progress so far: what has been accomplished, what remains to be done. If you have encountered problems, discuss them, and your plans to move forward. If you need help to make progress, please mention it.
- These reports need not be long, just a few sentences is fine.

Announcements: Time

- Time is going by quickly. You have 6 project weeks. We're in the middle of week 2 at this point. Is your project 1/3 complete?
- I'm going to be away March 27, 28, 30. At least two TA's will be in the lab that week.

Announcement: Midterm results



Midterm results

We will offer you the chance to do problem 3 (writing the code to make the MSP430 into a simple data acquisition system) again, as a take-home exam problem.

You'll have the chance to recoup 2/3 of the marks you lost on that question by turning in another solution.

Rules:
1) open notes – you can access the lecture notes and the MSP430 datasheets, but no other resources.
2) work individually – do not discuss with classmates
3) you can test the code and see if it compiles and runs, if you want (if so, describe what you did!)
4) Due: by end of day Wednesday March 8.

Marking. If you got 8/8 on this question, doing it again will get you nothing, but a redo will never subtract. So, if new score > old score, then:

new score = old score + (8-old score) x 2/3 **x** ((new score-old score)/(8-old score))

Eg if you got 2/8 on this question on the exam, then a perfect score on the re-do will bring you up to:
2 + (8-2) x 2/3 x ((8-2)/(8-2)) = 6/8.

Midterm 3

3. We want to program the MSP 430 to act as a data acquisition system. Pins P1.0 and P1.3 to P1.7 will be connected to various devices that supply analog voltages. Pins P2.0 to P2.5 will be connected to logic devices. The data acquisition system will allow a program on the host computer to either request that the MSP430 measure one of the P1.x analog voltages, or to measure all of the logic signals on P2.0-2.5.

The program on the MSP430 should:
a) configure pins P1.1 and P1.2 as a serial interface using the USCI module [already done in the skeleton code provided]
b) configure pins P1.0 and P1.3 to P1.7 as inputs (**without** pull-up/down resistors).
c) configure pins P2.0 to P2.5 as input **with** pull-ups enabled [for the P2 pins, use P2DIR, P2OUT, P2IN etc]
d) put the cpu to sleep until a byte is received over the serial interface from the host computer
e) When a byte is received:
- the received byte should be transmitted back to the host computer.
  [already done in the skeleton code]
-If the value of the received byte is 0, 3, 4, 5, 6 or 7, then the ADC should be turned on and configured to make a measurement on the corresponding port 1 pin. The measurement from the ADC should then be transmitted to the host computer (as two consecutive bytes: low byte then high byte). The ADC should then be turned off
- If the received byte is an 8, then the program should read the state of the port 2 logic pins, and transmit them as a single byte
- Any other received value should be ignored.

The skeleton code on the next page handles 9600 baud serial communication with the host computer, receiving bytes and echoing them back to the host. The example code we saw in the lab to do ADC measurements is provided in the reference materials for you to draw from.

Complete the program so that it will operate as described above, including useful comments where appropriate so that your code is understandable. You **do not** need to explain the code that is already there.

t(/8)

```
#include <msp430.h>
int main(void){
      WDTCTL = WDTPW + WDTHOLD;
      BCSCTL1 = CALBC1_1MHZ; // Set DCO to 1MHz
      DCOCTL = CALDCO_1MHZ; // Set DCO to 1MHz

      /* Configure hardware UART */
       P1SEL = BIT1 + BIT2 ; // P1.1 = RXD, P1.2=TXD
       P1SEL2 = BIT1 + BIT2 ; // P1.1 = RXD, P1.2=TXD
       UCA0CTL1 |= UCSSEL_2; // Use SMCLK
       UCA0BR0 = 104; // Set baud rate to 9600 with 1MHz clock (Data Sheet 15.3.13)
       UCA0BR1 = 0; // Set baud rate to 9600 with 1MHz clock
       UCA0MCTL = UCBRS0; // Modulation UCBRSx = 1
       UCA0CTL1 &= ~UCSWRST; // Initialize USCI state machine
       IE2 |= UCA0RXIE; // Enable USCI_A0 RX interrupt

/* serial set-up complete */




}
// this interrupt is called when a byte is received over the serial interface:
void __attribute__ ((interrupt(USCIAB0RX_VECTOR))) rx_vector(void){
      unsigned char inbyte;
      inbyte = UCA0RXBUF;//read received byte. This also resets the interrupt flag
      while(!(IFG2 & UCA0TXIFG)); // wait for any previous byte to be sent
      UCA0TXBUF = inbyte;  // echo back to host









}
```

A word about data sheets

- Beware of sections entitled “Absolute Maximum Ratings”
- These sections tell you about the most extreme conditions the component can be subjected to without being destroyed. These conditions are usually very far away from the optimal operating conditions! To find suitable operating conditions, there is often a table of Electrical Parameters – look for the conditions under which other parameters are measured.

WIEDNER HAUPTSTRASSE 76 1040 VIENNA AUSTRIA
TEL. +43 1 586 52 43 -0, FAX. -44, OFFICE@ROITHNER-LASER.COM

LED1200-series

TECHNICAL DATA

Infrared LED — InGaAsP

LED1200-series are InGaAsP LEDs mounted on a lead frame and encapsulated in various types of epoxy lens, which offers different design settings.
On forward bias, it emits a high power radiation of typical 5 mW at a peak wavelength at 1200 nm.

Specifications

- Structure: InGaAsP
- Peak Wavelength: typ. 1200 nm
- Optical Ouput Power: typ. 5 mW
- Resin Material: Epoxy resin
- Solder: Lead free

Absolute Maximum Ratings (T_a=25°C)

Type	Symbol	Value	Unit
Power Dissipation	P_D	140	mW
Forward Current	I_F	100	mA
Pulse Forward Current	I_{FP}	1000	mA
Reverse Voltage	V_R	5	V
Operating Temperature	T_{OP}	-40 … +85	°C
Storage Temperature	T_{STG}	-40 … +100	°C
Soldering Temperature (for 5 sec.)	T_{SOL}	265	°C

Electro-Optical Characteristics (T_a=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Forward Voltage	V_F	I_F = 50 mA	-	1.1	1.5	V
Reverse Current	I_R	V_R = 5 V	-	-	10	µA
Radiated Power	P_O	I_F = 50 mA	3	5	-	mW
Peak Wavelength	λ_P	I_F = 50 mA	1150	1200	1250	nm
Half Width	$\Delta\lambda$	I_F = 50 mA	-	80	-	nm
Rise Time	t_r	I_F = 50 mA	-	10	-	ns
Fall Time	t_f	I_F = 50 mA	-	10	-	ns

Single supply amplifier

- When working with microcontrollers it is often convenient to have an amplifier that can be powered from 0/5V or 0/3.3V rather than +/-15V.
- Previous circuits need some modifications: (a) need to reference inputs from the supply midpoint. (b) often want to AC couple the input.

  TI has some nice documents:

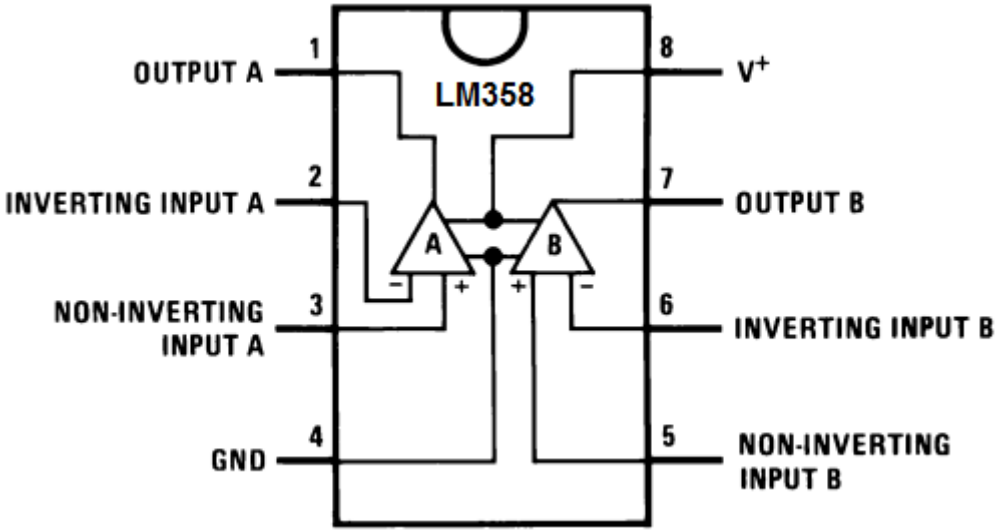
https://courses.cit.cornell.edu/bionb440/datasheets/SingleSupply.pdf

www.ti.com/lit/ml/sloa091/sloa091.pdf

www.ti.com/lit/an/sloa030a/sloa030a.pdf

Single supply amplifier

- LM358. Dual, single supply.
- V+ = 5V (pin 8), V- = 0V (pin 4).
- Outputs can swing from ~0 V to ~ 3.5 V.



General Idea, but awkward choices:

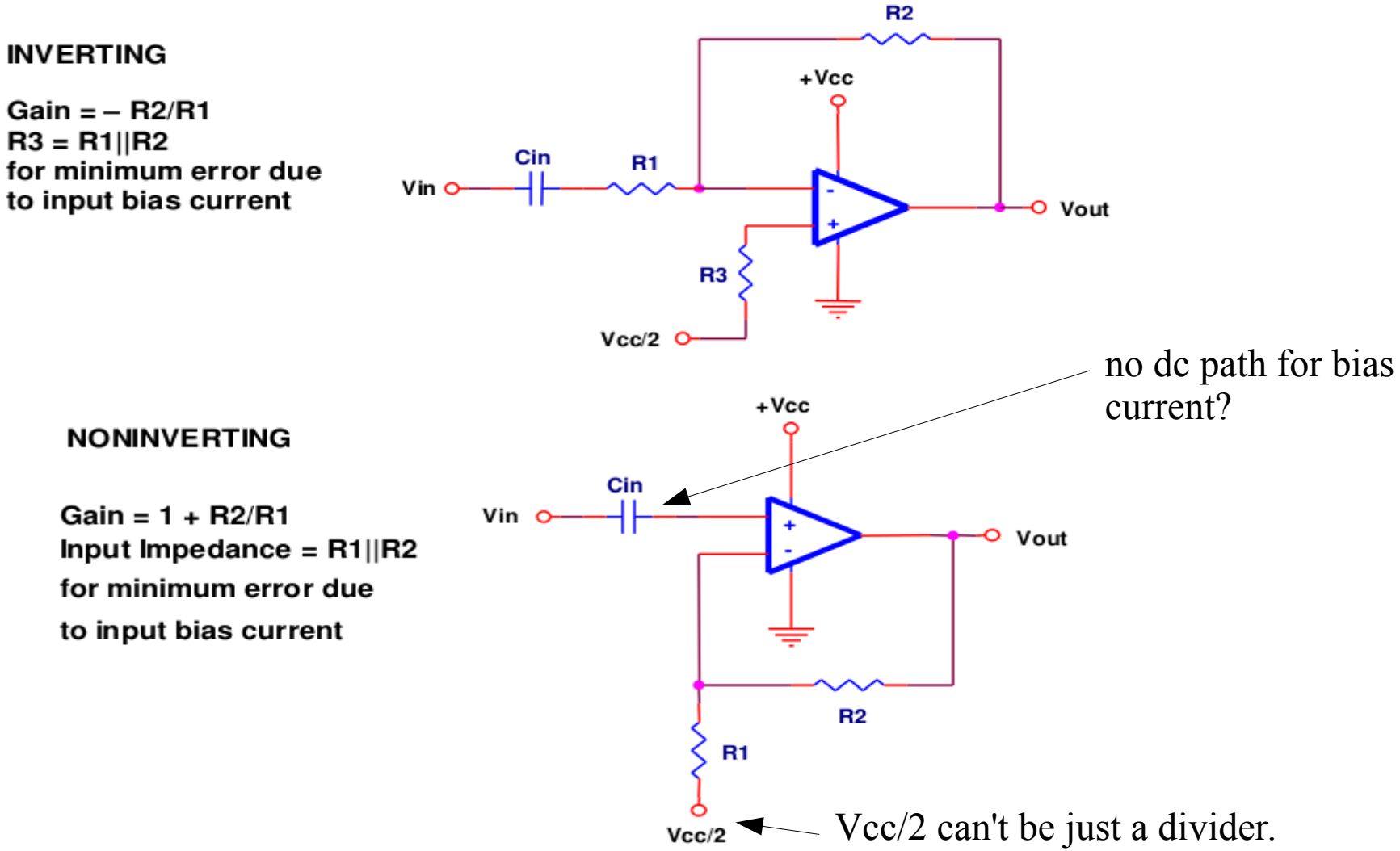
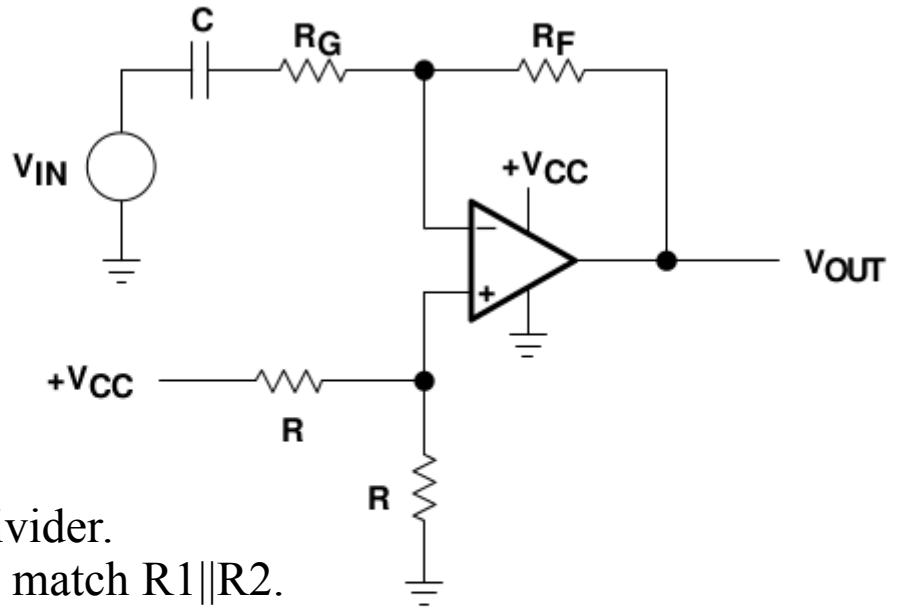


Figure 3. AC-Coupled Gain Stages

https://courses.cit.cornell.edu/bionb440/datasheets/SingleSupply.pdf

- Another choice for inverting AC amplifier:

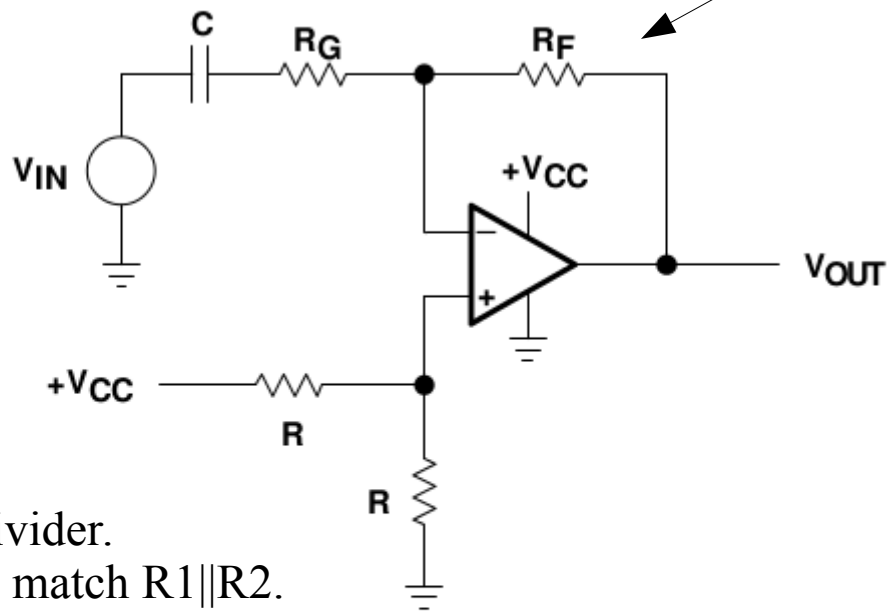


Make V+ here = 1.67 V with a divider.
Choose the divider impedance to match R1||R2.
The two resistors in the divider aren't the same.

Figure A–22. Inverting AC Amplifier

- Another choice for inverting AC amplifier:

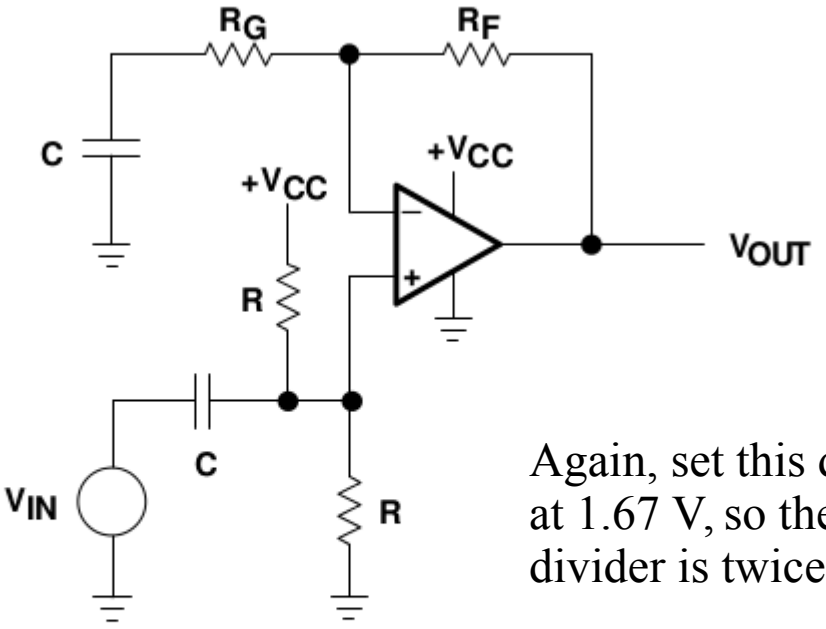
add a capacitor in parallel with R_F to low-pass filter



Make V+ here = 1.67 V with a divider.
Choose the divider impedance to match R1||R2.
The two resistors in the divider aren't the same.

Figure A–22. Inverting AC Amplifier

- And non-inverting amplifier:

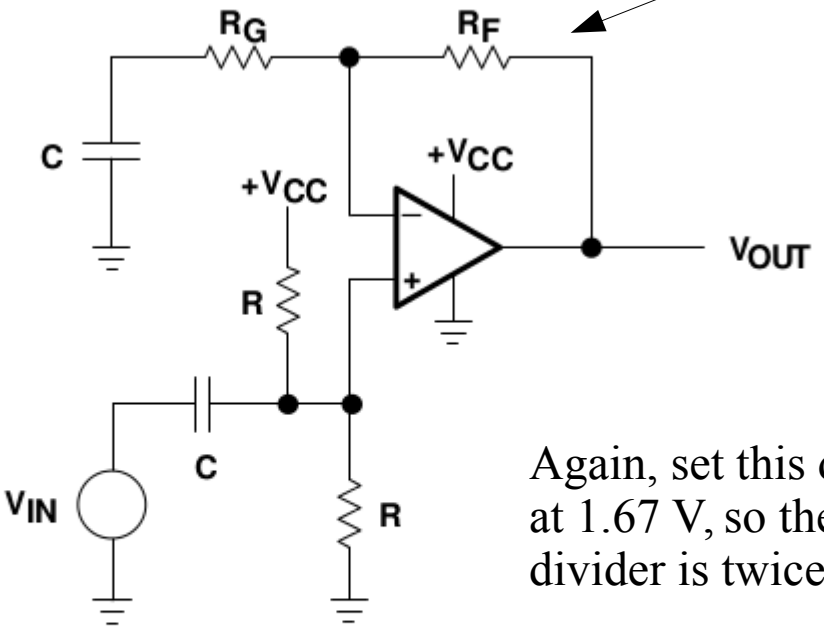


Again, set this divider to V+ idles at 1.67 V, so the upper resistor on the divider is twice as big as the lower.

Figure A–23. Noninverting AC Amplifier

- And non-inverting amplifier:

add a capacitor
in parallel with R_F
to low-pass filter



Again, set this divider to V+ idles at 1.67 V, so the upper resistor on the divider is twice as big as the lower.

Figure A–23. Noninverting AC Amplifier

2nd order filters:

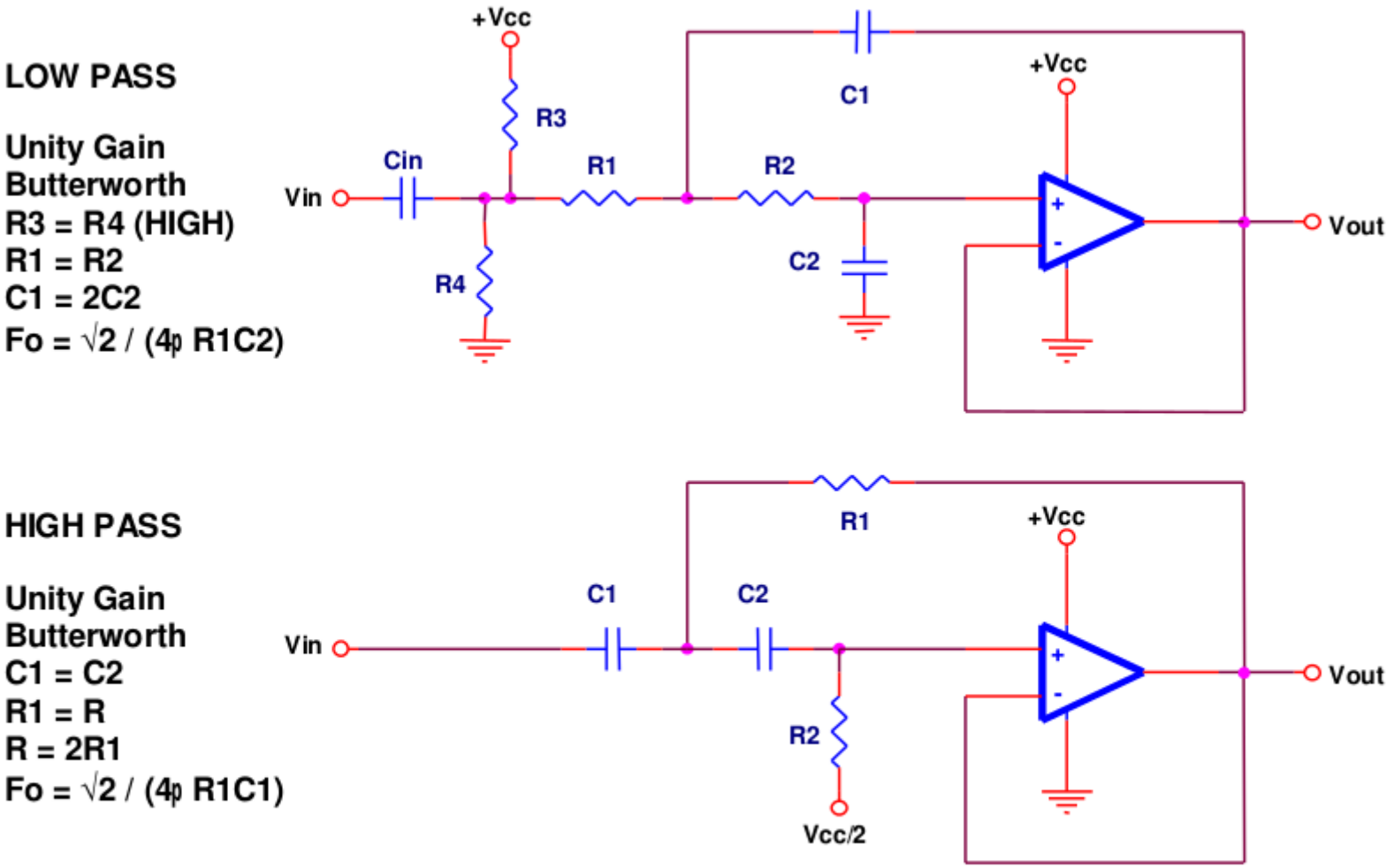


Figure 16. Sallen-Key Low- and High-Pass Filter Topologies

See also, Horowitz and Hill, The Art of Electronics

See also, Horowitz and Hill, The Art of Electronics

2nd order filters:

Bandpass filter designer:
https://web.archive.org/web/20070209043913/http://www.captain.at/electronics/active-filter/

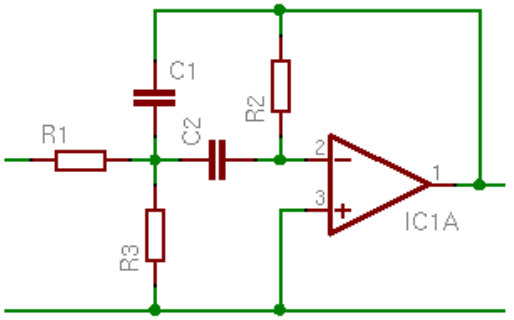
Active Filter Calculator - Bandpass with OpAmp Designer in Javascript

A simple tool for designing active filters using voltage-feedback opamps.

Type in the center frequency (f_c), bandwidth (B), gain (A) and choose a value for the capacitors (C1, C2 = C; 10nF is a good value to start) and click on "Calculate".

Light-grey fields are input fields - dark-grey fields are output fields for the calculated values.

f_c =	Hz	Q =	
B =	Hz	R1 =	Ω
A =		R2 =	Ω
C =	nF	R3 =	Ω
Calculate			



Reverse engineer the active filter:
Use the "Resistor Series lookup" tool below to get values for available resistors and calculate the actual center frequency, bandwidth, Q factor and gain.

C =		nF
R1 =		Ω
R2 =		Ω
R3 =		Ω
Calculate		
Q =		
f_c =		Hz
B =		Hz
A =		

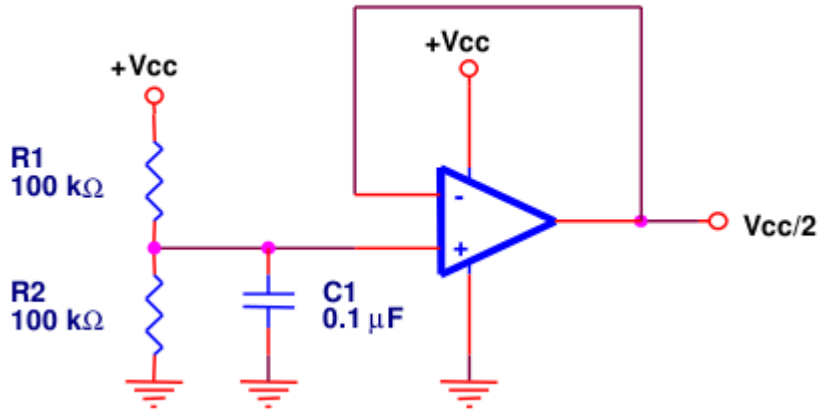
Resistor Series lookup
Type in R in Ohm, select the series and click "Suggest"

R =		Ω
◉ E6 ○ E12 ○ E24 ○ E48		
Suggest		
R_{lower} =		Ω
R_{upper}		Ω

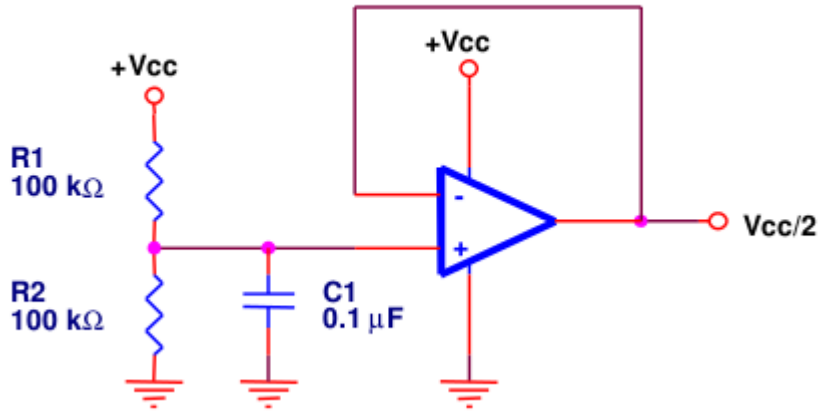
Provide center frequency, bandwidth, gain, and a value for the capacitors, the designer will supply resistor values. If they are ridiculously large or small, try a larger or smaller C. Keep bandwidth and gain reasonably (eg B ~ 0.05 to 0.20 f_c, and A < 20.

To use with single supply op-amp, connect V+ to a divider at 1.67V. But leave R3 to ground.

- If you need to generate Vcc/2 (which is the op-amp "ground") use a second op-amp:



- If you need to generate Vcc/2 (which is the op-amp "ground") use a second op-amp:



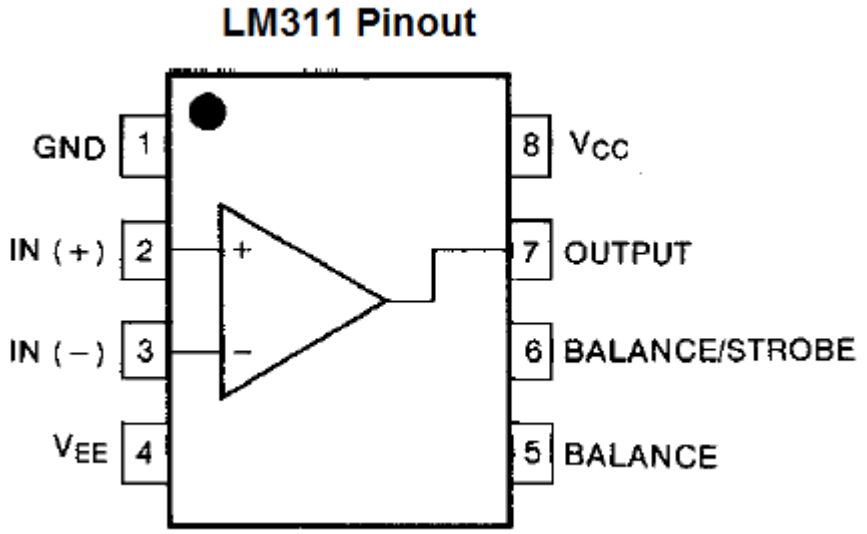
If Vcc = 5V,
make R_1=200k, R_2= 100k,
puts output at:
1.67 V.

Comparators

Comparators

Basic device function: compare two voltages, indicate which is greater.

But also useful for:
- logic level shifting,
- threshold detection/ generating square waves
- driving the P-channel mosfet or pnp transistor on H-bridges
- turning a logic output into a 'tri-state' output.

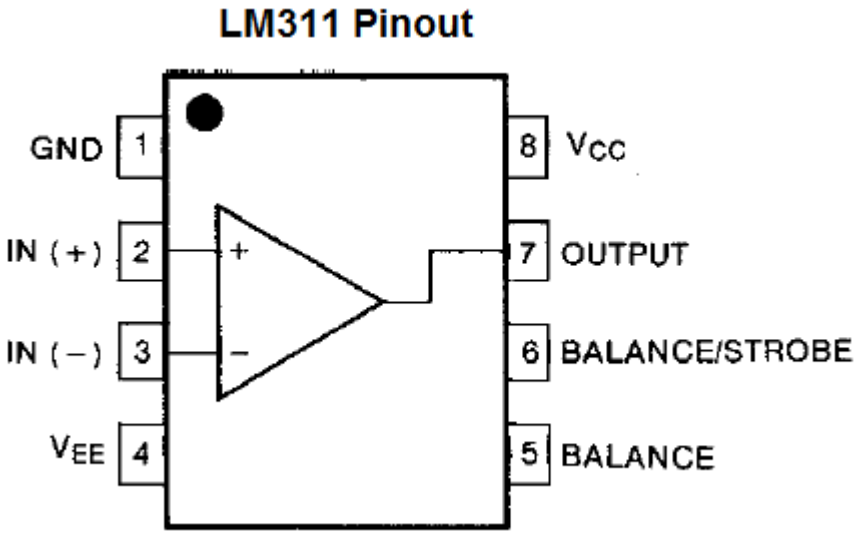


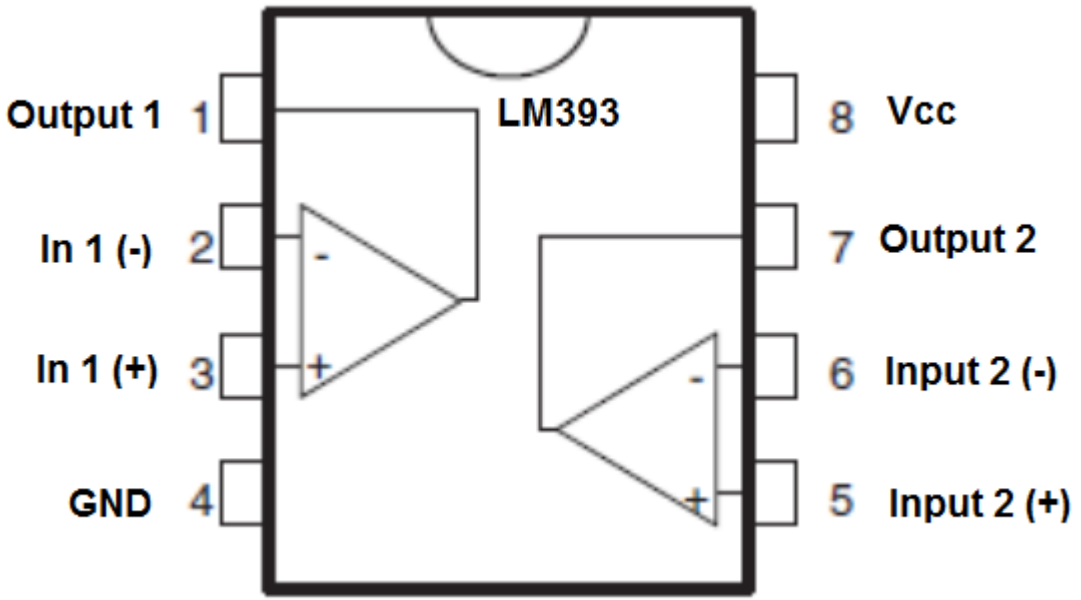
Comparators

V_{CC}, V_{EE} - +, - supplies. The inputs must stay between the supply voltages. Can be +/-15V or +5/0.

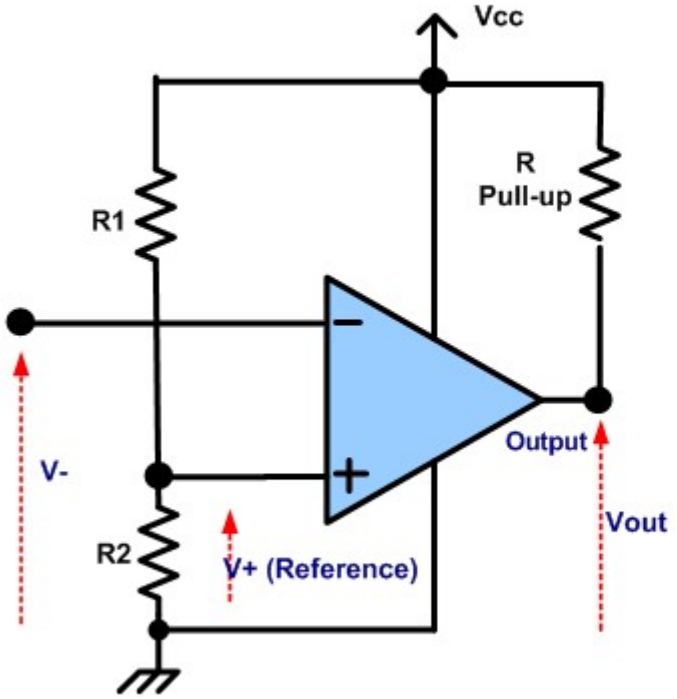
When V- > V+, then the output is connected to GND. When V+ > V-, the output floats.

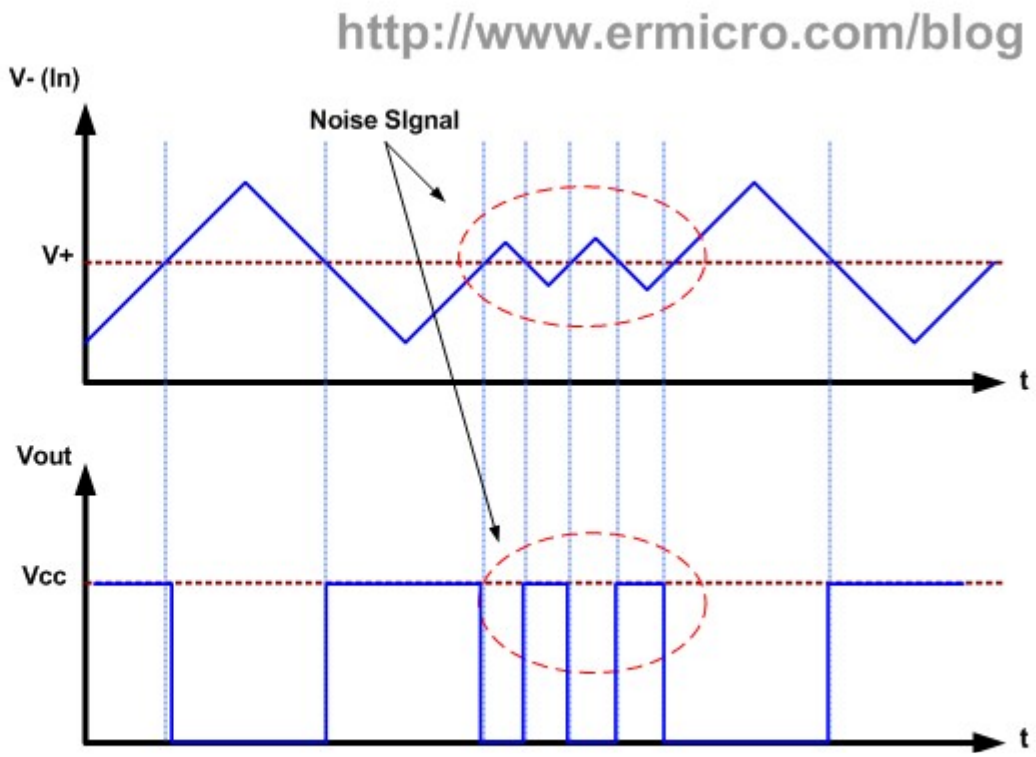
Balance: used to trim internal V+ vs V- offsets. Not usually needed.
Strobe: pull to ground to disable comparator.





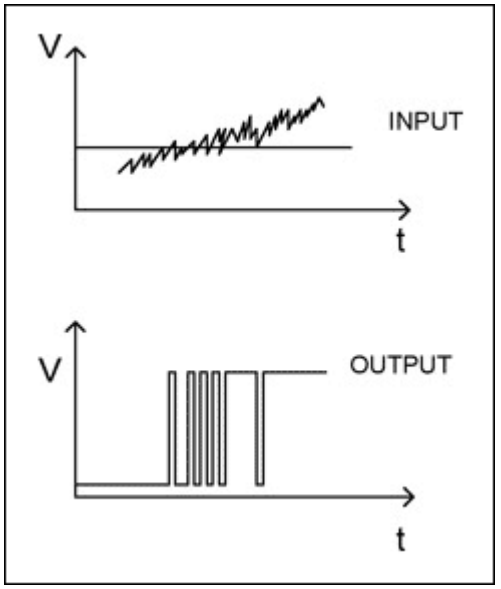
Comparators





Comparators

Noisy signals:



http://hades.mech.northwestern.edu/index.php/Comparators

Comparators

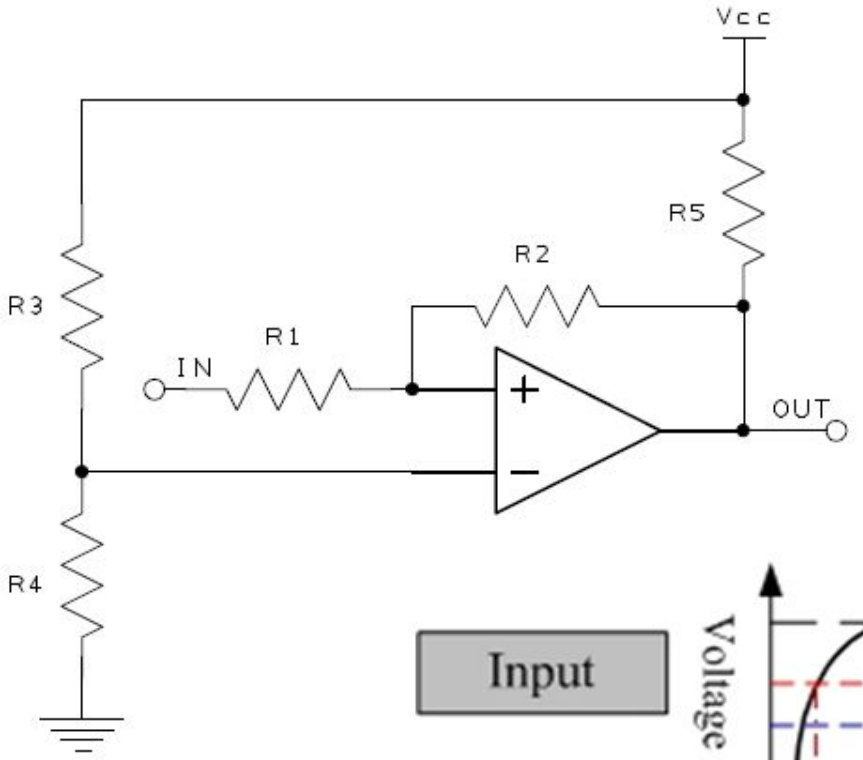
Hysteresis for noise immunity: add positive feedback.

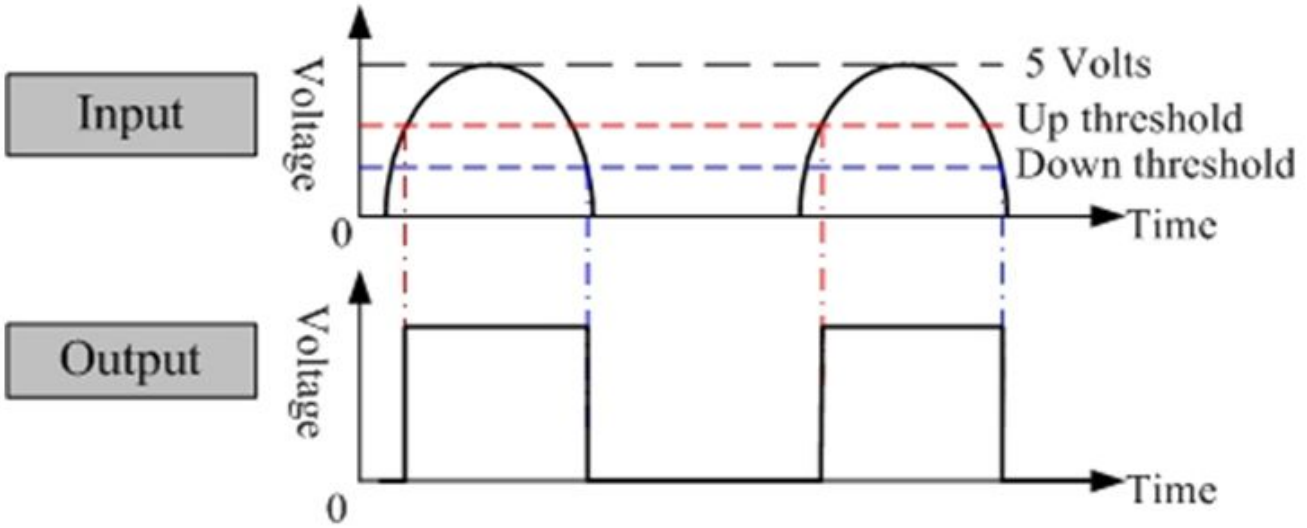
$$V_{ref} = V_{cc}\left(\frac{R4}{R4+R3}\right)$$

$$V_U = V_{ref}\left(\frac{R1+R2}{R2}\right)$$

$$V_L = \frac{V_{ref}\,(R1+R2) - V_{cc}\,(R1)}{R2}$$

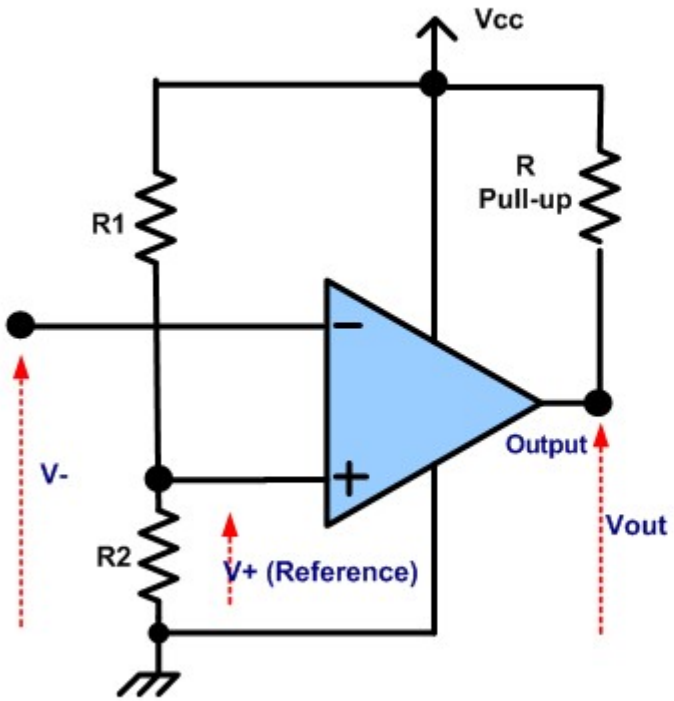
(Assumes $R_5 << R_2$)





http://hades.mech.northwestern.edu/index.php/Comparators

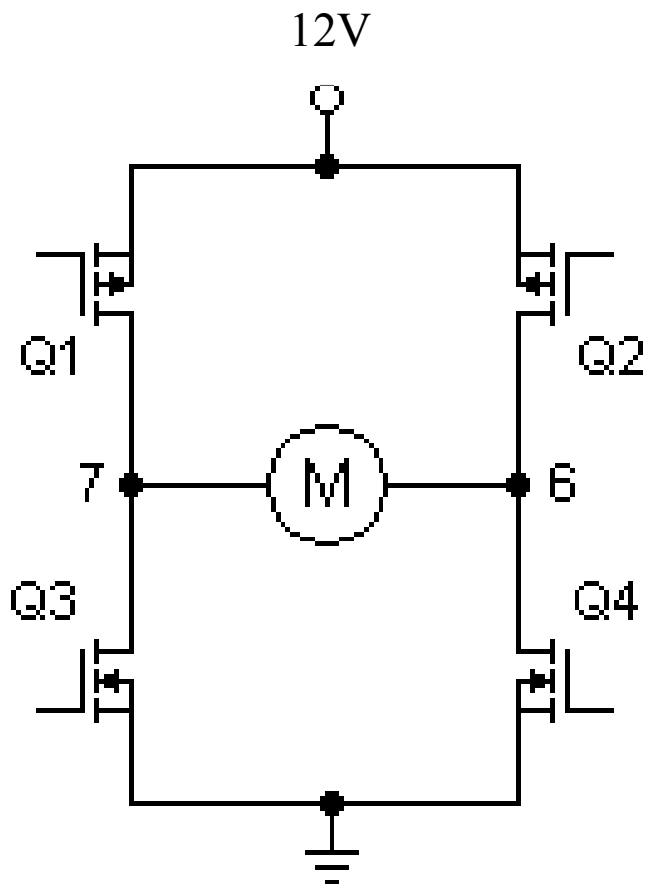
Comparators



The pull-up doesn't have to be connected to the same supply voltage as the comparator supply, it can be higher or lower. This makes the comparator output very flexible for level shifting!

Comparators

Example: Level shifting:

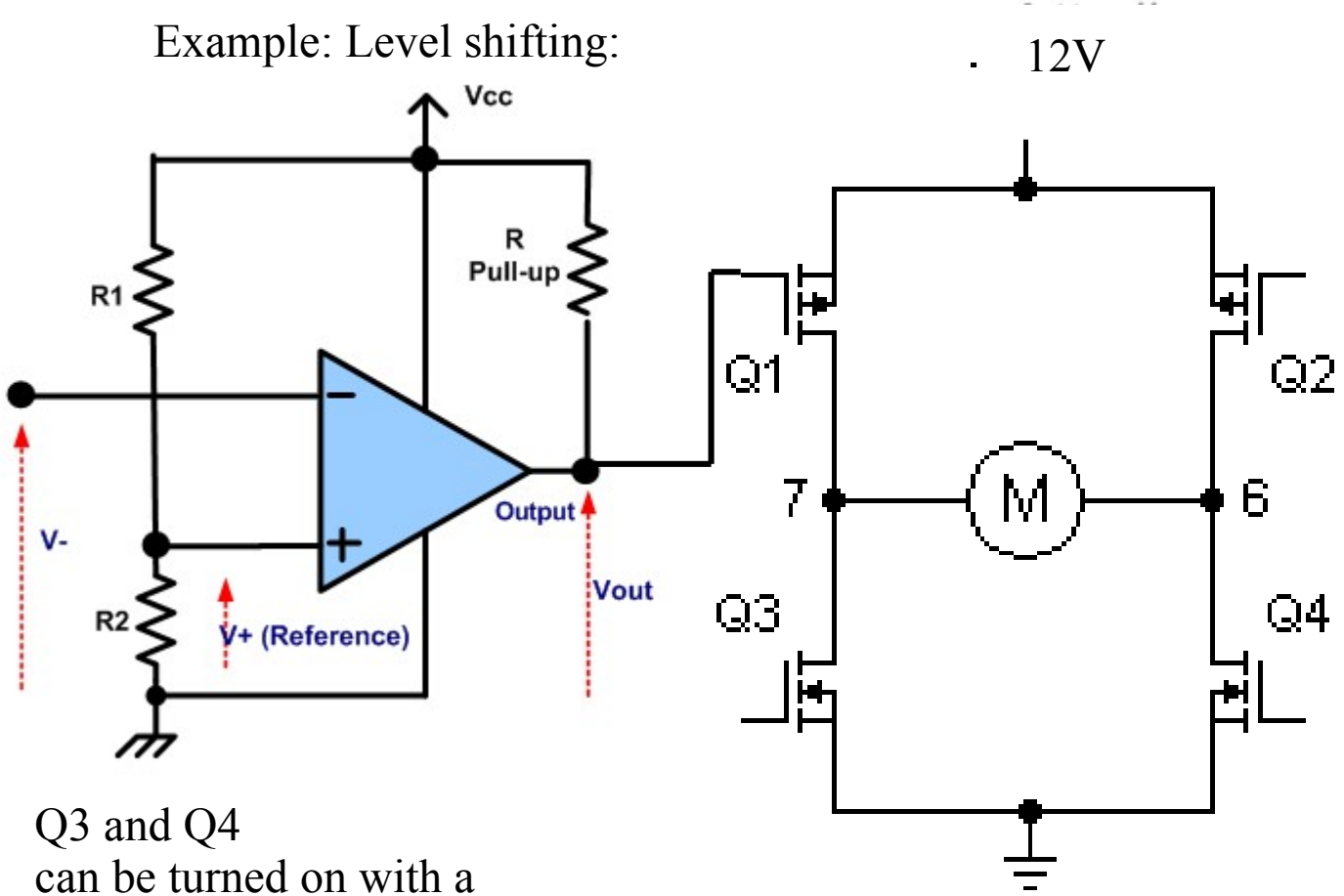


Q3 and Q4
can be turned on with a
5V logic device, off at 0.

But Q1 and Q2 need to be up at 12V to be turned off, then pulled down to turn on.

Comparators

Example: Level shifting:



Q3 and Q4
can be turned on with a
5V logic device, off at 0.

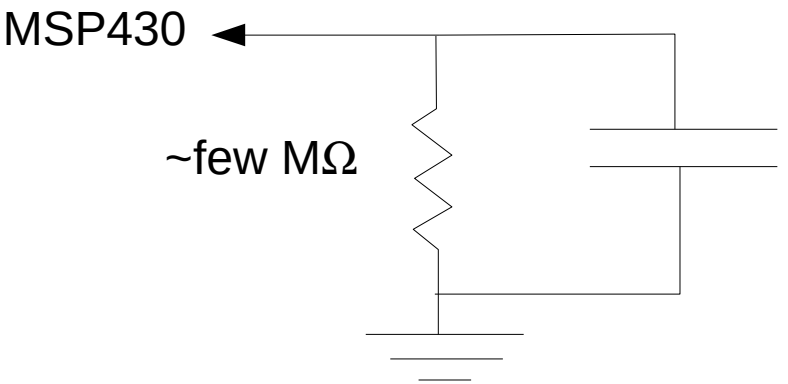
But Q1 and Q2 need to be up at 12V to be turned off, then pulled down to turn on.

Capacitance Measurements

- reference document from TI

  http://www.ti.com/lit/an/slaa379/slaa379.pdf

  Simplest:



- set the pin as an output, and set it H
- then set as input, and time how long it takes to discharge to read as low.

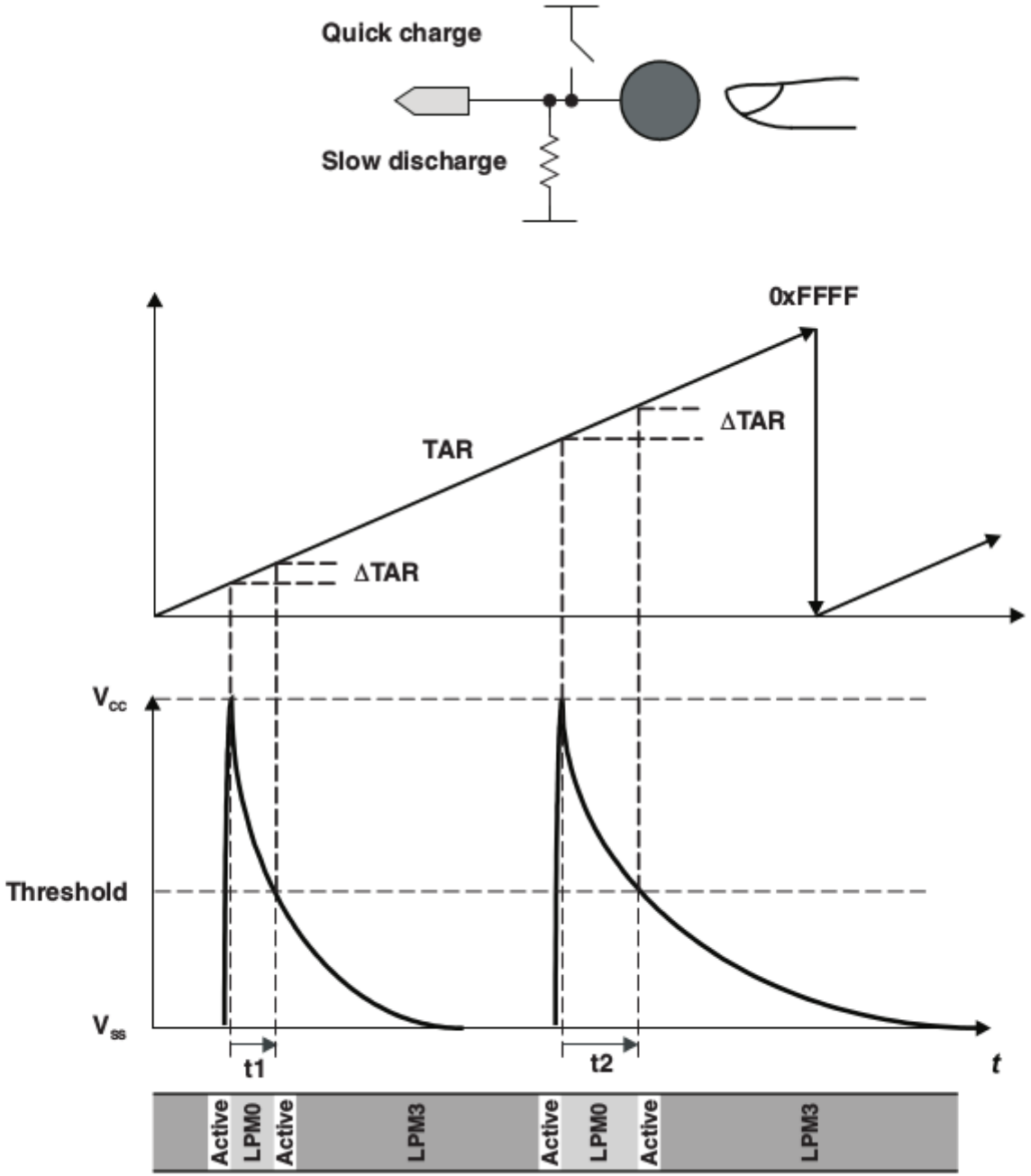


Figure 2. Charge-Discharge Sequence

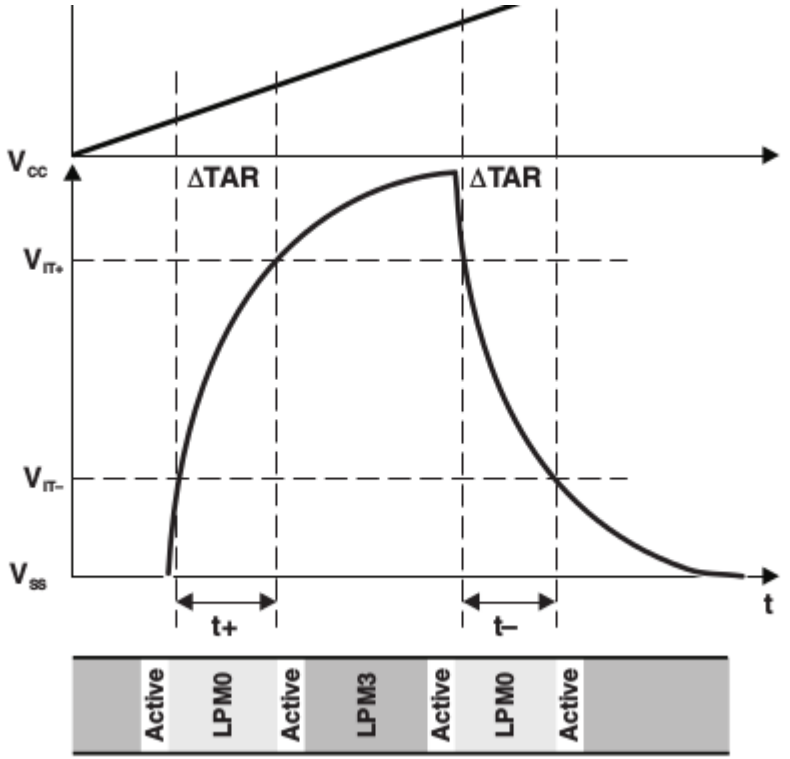


Figure 3. Measurement Cycle for Improved Noise Rejection

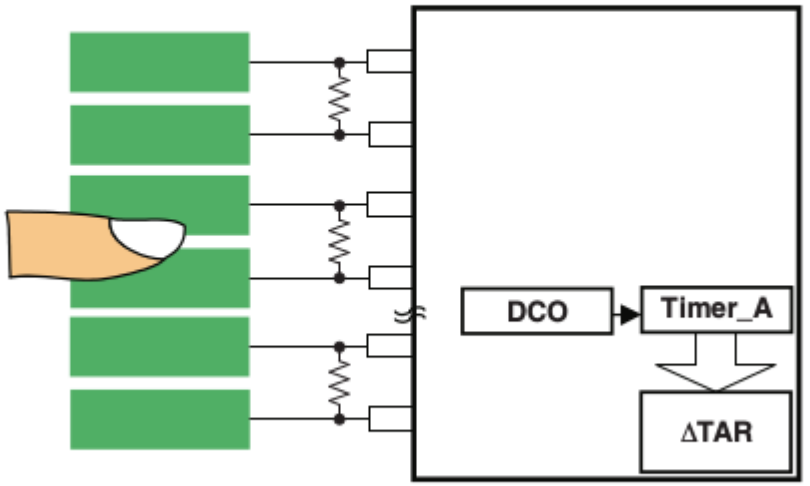


Figure 4. Multi-Sensor Charge/Discharge Configuration

Better noise suppression

Software low-pass filter:

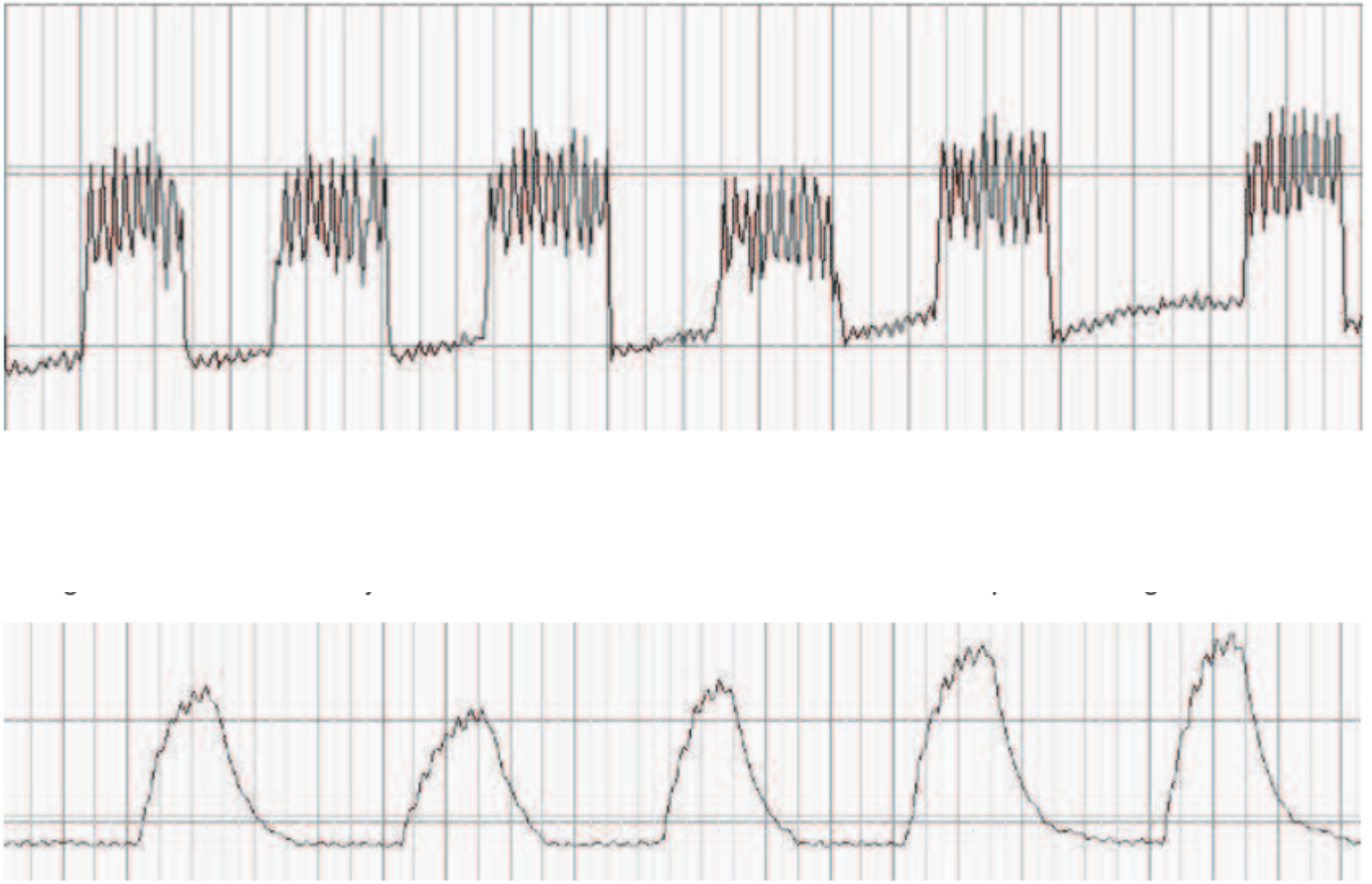


Figure 7. Oscilloscope With IIR Filter

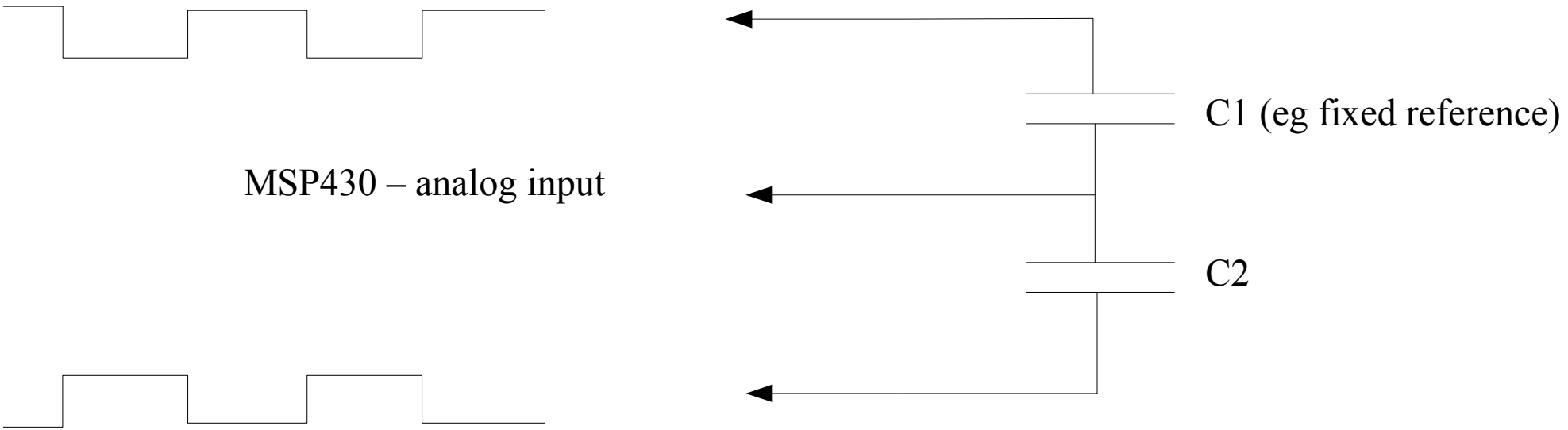
Better noise suppression

Software low-pass filter:

```
int current,filter;

// make measurement in here:

current = most recent measurement


filter = (1-K) * current + K*filter; // filter response same as simple RC low pass filter

// better implemented as, eg:
filter = (15*filter +current)/16;

// then output filter value.
```

Better noise suppression

Differential capacitance measurement:



Pin Oscillator

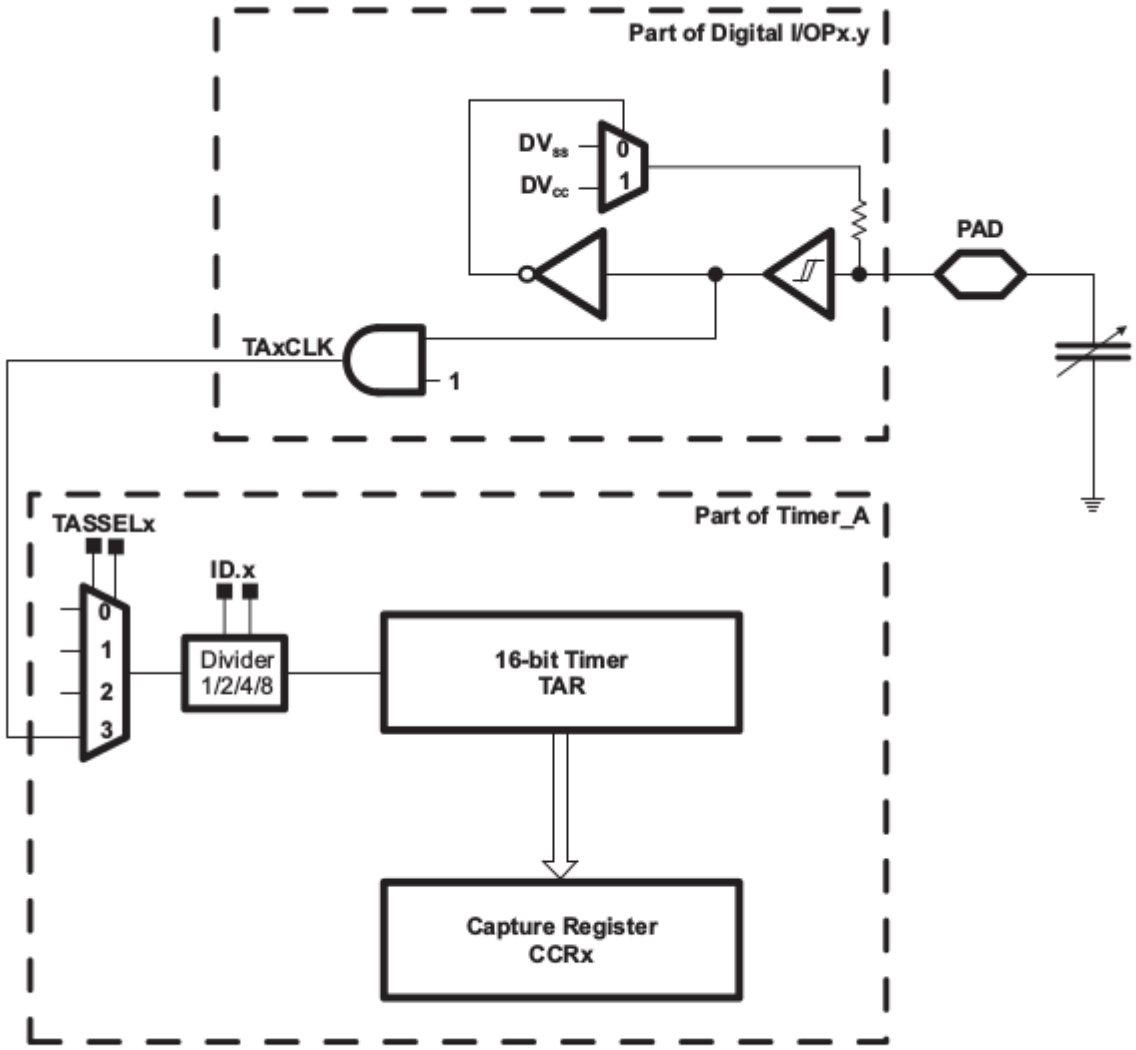
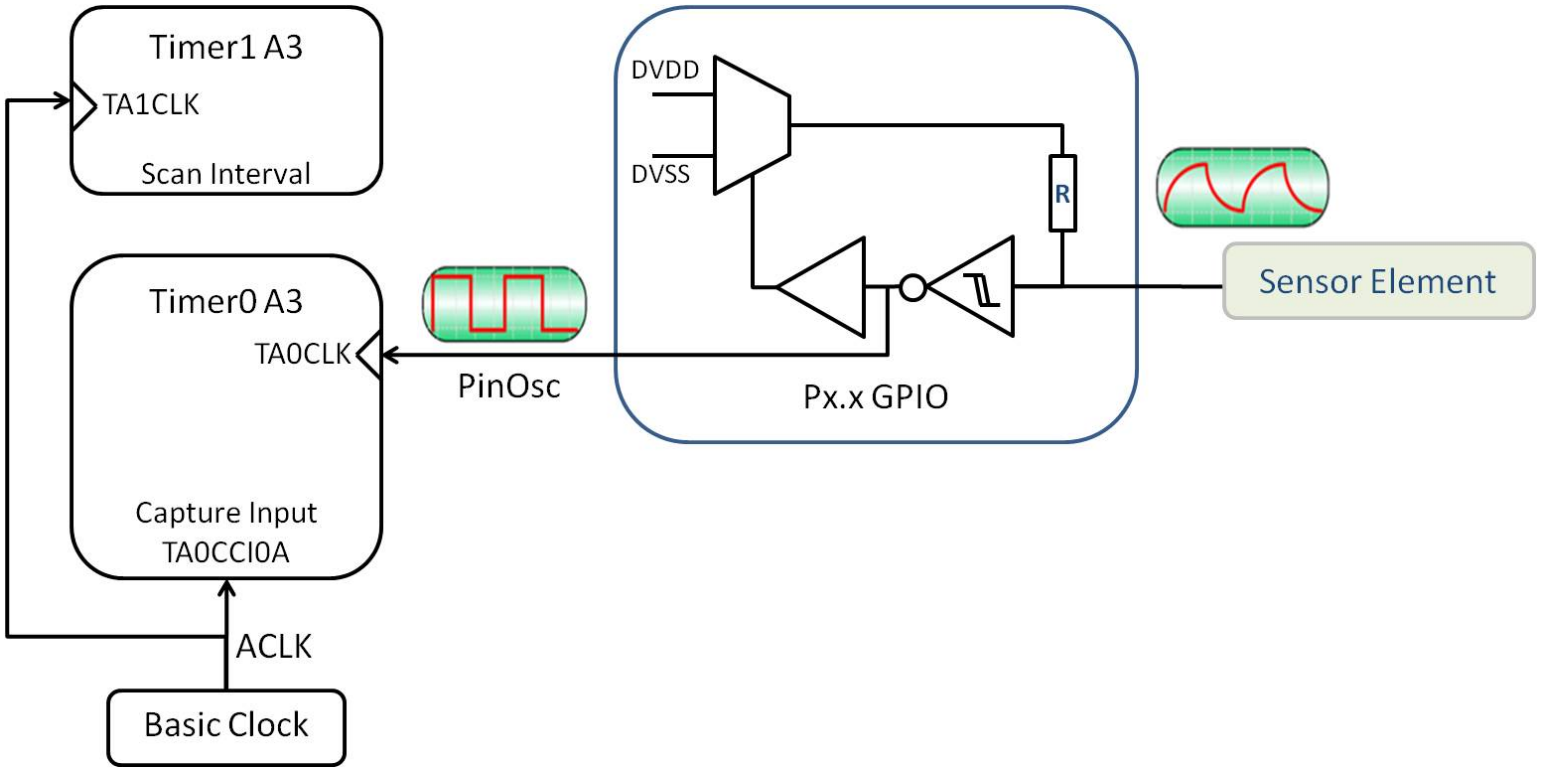


Figure 8-1. Example Circuitry and Configuration using the Pin Oscillator

See sec 8.2.6 in slau144 and Table 16 in slas735.
The oscillator rate depends on the capacitance applied to the pin.

The oscillator is used to drive a timer. By comparing the timer rate to the rate of some other timer, you can determine the capacitance.

See http://processors.wiki.ti.com/index.php/MSP430_Low_Cost_PinOsc_Capacitive_Touch_Overview



- Here the pin oscillator runs quickly (~MHz), and ACLK is configured to run slowly. Every ACLK cycle triggers a Capture event that stores the pin oscillator count.

- Frequency measurement applications are very similar: configure a timer to run quickly (eq SMCLK at 1MHz, then trigger CCR captures from the audio (or whatever else) you want to measure the frequency of.
- TACTL: source = SMCLK, up mode, eg
  TACTL = TASSEL_2 | MC+2;
- TACCTL0: enable input capture: CAP, capture mode: rising edge, falling edge, or both. CCIE: trigger interrupts. eg:
  TACCTL0 = CM_1 | CAP | CCIE;
- Interrupt handler remembers the previous timer value, subtracts from most recent timer value to measure period. eg:

```
{
static unsigned int last;
current = TACCR0;
period = current – last;

// maybe wake up cpu here to communicate the period?
// might want to check for timer overflows?
}
```

12.3.1 TACTL, Timer_A Control Register

15	14	13	12	11	10	9	8
Unused						TASSELx	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
IDx		MCx		Unused	TACLR	TAIE	TAIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Unused	Bits 15-10	Unused
TASSELx	Bits 9-8	Timer_A clock source select 00 TACLK 01 ACLK 10 SMCLK 11 INCLK (INCLK is device-specific and is often assigned to the inverted TBCLK) (see the device-specific data sheet)
IDx	Bits 7-6	Input divider. These bits select the divider for the input clock. 00 /1 01 /2 10 /4 11 /8
MCx	Bits 5-4	Mode control. Setting MCx = 00h when Timer_A is not in use conserves power. 00 Stop mode: the timer is halted. 01 Up mode: the timer counts up to TACCR0. 10 Continuous mode: the timer counts up to 0FFFFh. 11 Up/down mode: the timer counts up to TACCR0 then down to 0000h.
Unused	Bit 3	Unused
TACLR	Bit 2	Timer_A clear. Setting this bit resets TAR, the clock divider, and the count direction. The TACLR bit is automatically reset and is always read as zero.
TAIE	Bit 1	Timer_A interrupt enable. This bit enables the TAIFG interrupt request. 0 Interrupt disabled 1 Interrupt enabled
TAIFG	Bit 0	Timer_A interrupt flag 0 No interrupt pending 1 Interrupt pending

12.3.4 TACCTLx, Capture/Compare Control Register

15	14	13	12	11	10	9	8
CMx		**CCISx**		**SCS**	**SCCI**	**Unused**	**CAP**
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	r0	rw-(0)
7	6	5	4	3	2	1	0
OUTMODx			**CCIE**	**CCI**	**OUT**	**COV**	**CCIFG**
rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	rw-(0)	rw-(0)	rw-(0)

CMx Bit 15-14 Capture mode
- 00 No capture
- 01 Capture on rising edge
- 10 Capture on falling edge
- 11 Capture on both rising and falling edges

CCISx Bit 13-12 Capture/compare input select. These bits select the TACCRx input signal. See the device-specific data sheet for specific signal connections.
- 00 CCIxA
- 01 CCIxB
- 10 GND
- 11 V_{CC}

SCS Bit 11 Synchronize capture source. This bit is used to synchronize the capture input signal with the timer clock.
- 0 Asynchronous capture
- 1 Synchronous capture

SCCI Bit 10 Synchronized capture/compare input. The selected CCI input signal is latched with the EQUx signal and can be read via this bit

Unused Bit 9 Unused. Read only. Always read as 0.

CAP Bit 8 Capture mode
- 0 Compare mode
- 1 Capture mode

OUTMODx Bits 7-5 Output mode. Modes 2, 3, 6, and 7 are not useful for TACCR0, because EQUx = EQU0.
- 000 OUT bit value
- 001 Set
- 010 Toggle/reset
- 011 Set/reset
- 100 Toggle
- 101 Reset
- 110 Toggle/set
- 111 Reset/set

CCIE Bit 4 Capture/compare interrupt enable. This bit enables the interrupt request of the corresponding CCIFG flag.
- 0 Interrupt disabled
- 1 Interrupt enabled

CCI Bit 3 Capture/compare input. The selected input signal can be read by this bit.

OUT Bit 2 Output. For output mode 0, this bit directly controls the state of the output.
- 0 Output low
- 1 Output high

COV Bit 1 Capture overflow. This bit indicates a capture overflow occurred. COV must be reset with software.
- 0 No capture overflow occurred
- 1 Capture overflow occurred

CCIFG Bit 0 Capture/compare interrupt flag
- 0 No interrupt pending
- 1 Interrupt pending

Powering your project

Powering your project

Easiest, if it works:

- Launchpad from your computer
- the board/external circuitry with the wall wart we've provided.
- any higher current devices (eg motors) from the bench supply.

Powering your project

- For a 'mobile' project you'll need batteries
- Other projects may need a DC supply with higher current or voltage capacity than the brick.

DC power supplies

- DC supplies come in two general flavours:

  Switching and Linear

- The difference between these is in the internal structure of the supply. Switching supplies tend to be smaller/lighter/cheaper/more efficient than linear, but can introduce noise (10's to 100's of kHz).

Wall Warts

- **Most** wall warts sold with consumer electronics are DC, switching, unregulated.
- The voltage only matches the specified output voltage when the current draw is near to the specified current capability. Lower current draw yields higher voltage, may be as much as twice the specified voltage!
- For driving motors, that may be ok, but for powering logic or amplifier circuits, you'll need to regulate wall wart outputs

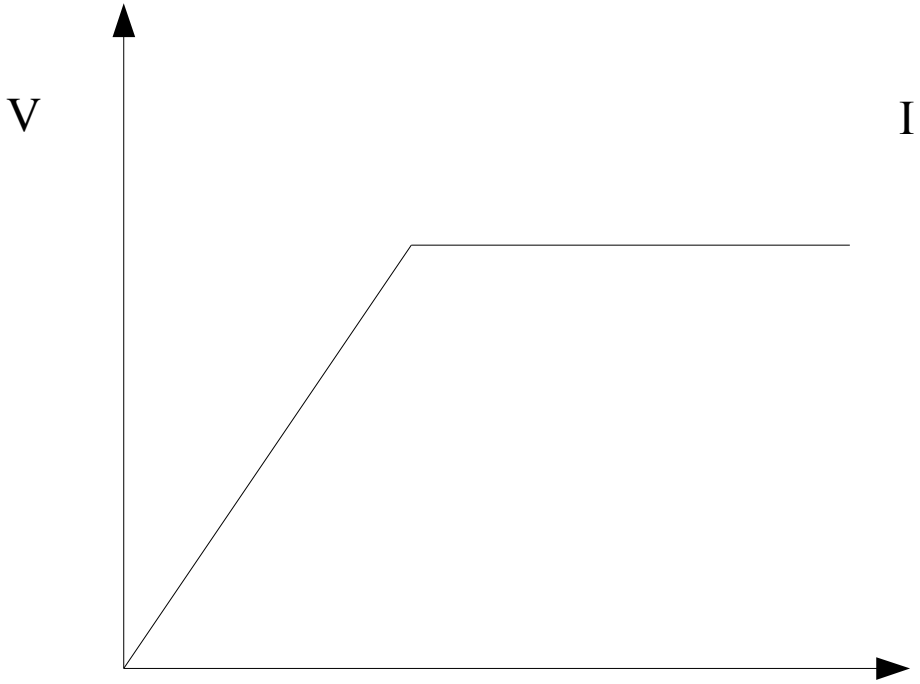
Wall Warts

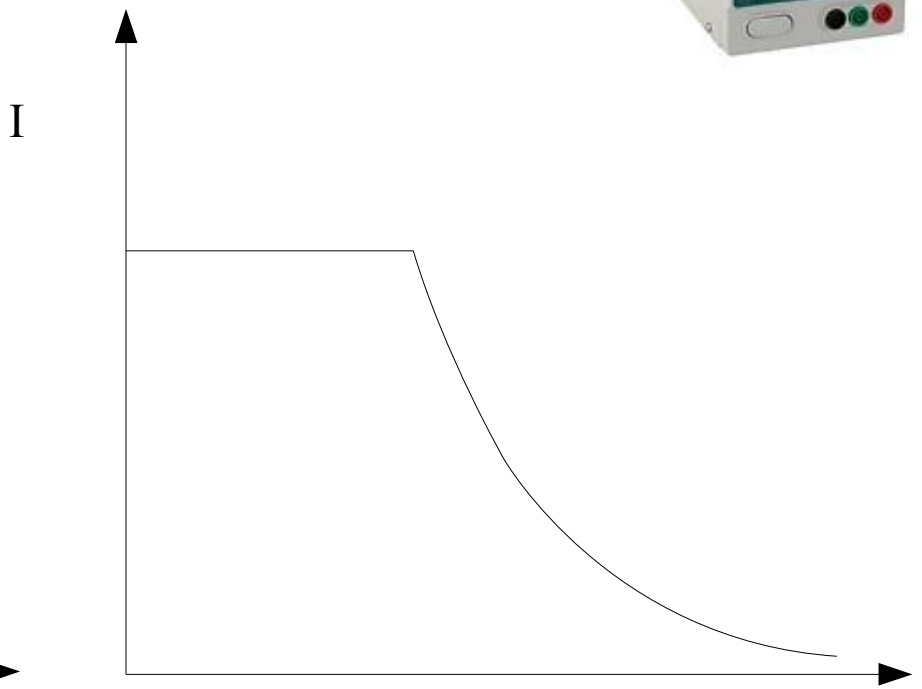
- Wall warts can be found that are linear, or AC, and/or are regulated. Often have to test to see if it is regulated or not.
- Newer wall warts with USB connections generally are regulated at 5V

Bench/Lab supplies

- Almost always linear
- Expensive
- Usually have voltage and current regulation
  So that you can specify a maximum voltage and a maximum current. With no load, the supply will raise its output voltage to the voltage setting. As the current draw is increased, the supply will maintain the set voltage until the current hits the current limit. At that point the voltage will drop and the current will be maintained.

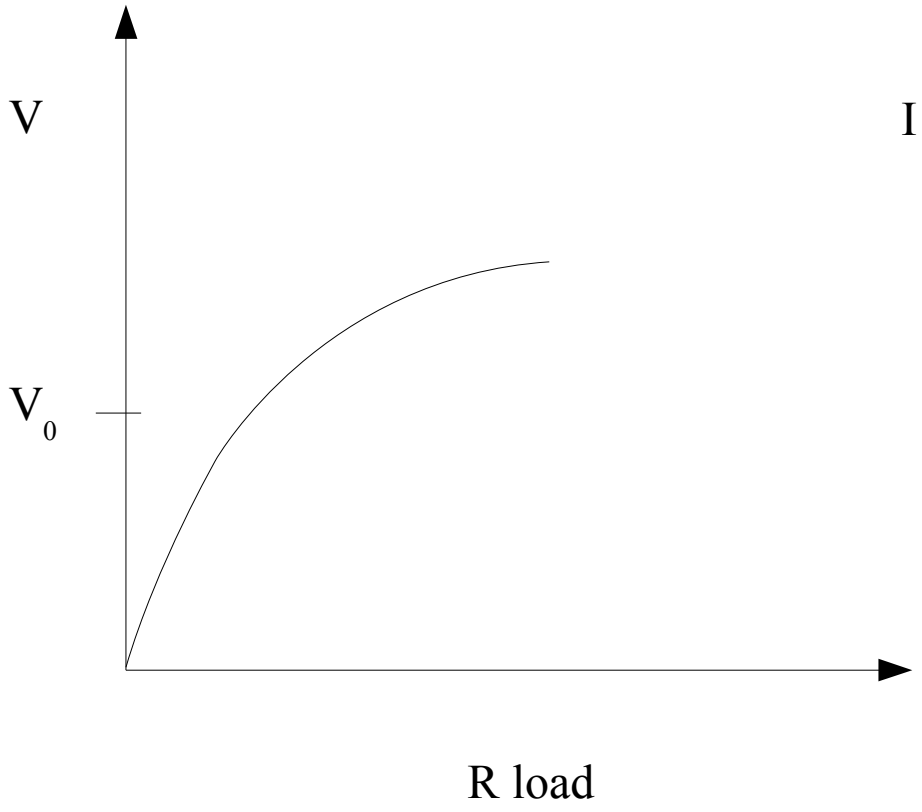
Bench/Lab supplies

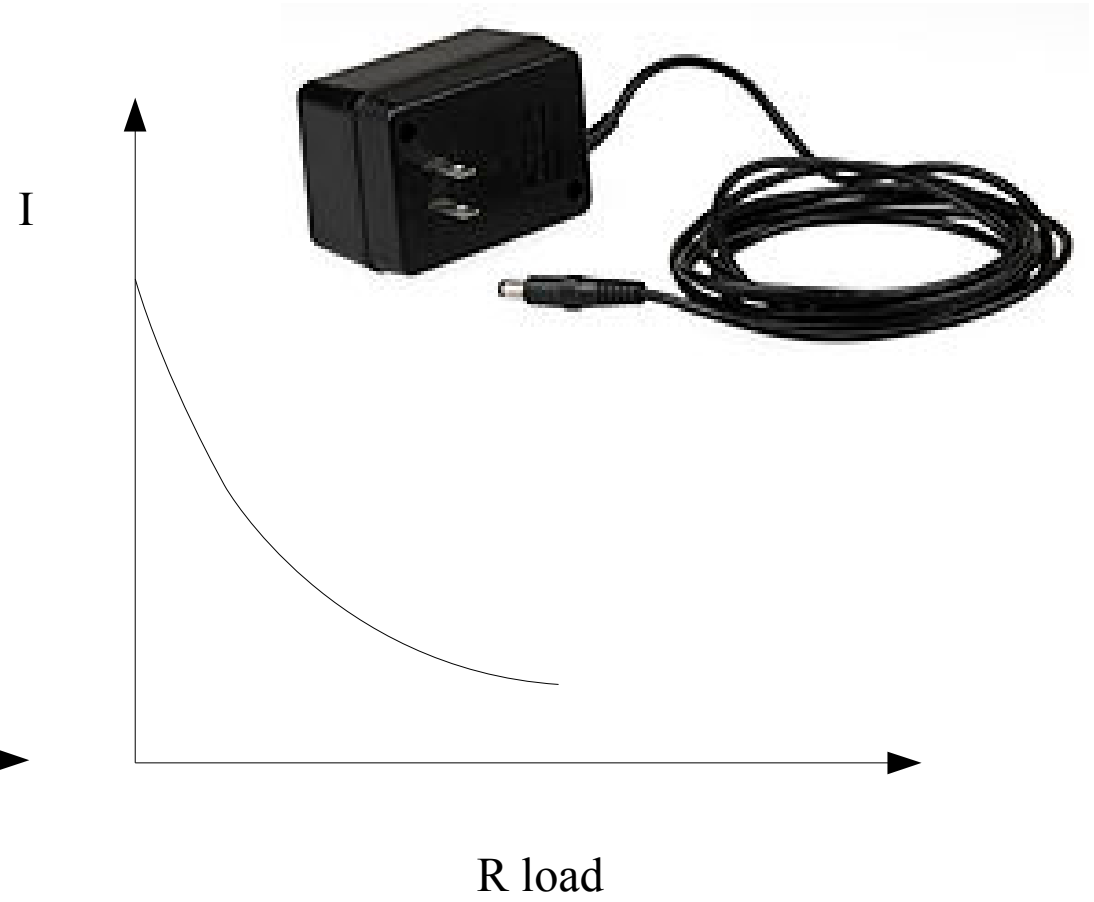




Wall Warts

Most unregulated wall warts, would look more like:





Batteries

- Many sizes/shapes/chemistries:
- Lead-acid - commonly available in 6V/12V. High power. Heavy, rechargeable.
- Lithium. Rechargeable or not. Rechargeables are a little tricky to use – must not overcharge or undercharge. Light weight.
- alkaline (AA and friends)
- Ni-MH/NiCd – easiest rechargeables to use
- coin cells/specialty (eg PX28L 6V camera battery)

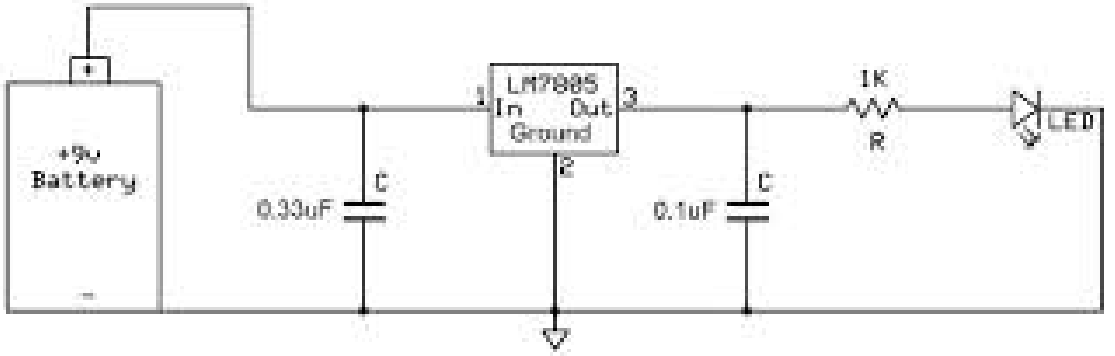
Batteries

- For most battery chemistries, the voltage changes as the battery is discharged. Eg alkalines start off ~ 1.5V, but discharge to ~ 1.0V.
- Many batteries can supply very high peak current – A fresh D battery can supply ~ 10A for a short period! Lead acid batteries can supply 100's of A. Due respect is required. Short circuit protection and possibly reverse connection protection should be considered.
- Like most other electronic components, batteries have data sheets with lots of useful information on them!

Voltage Regulation

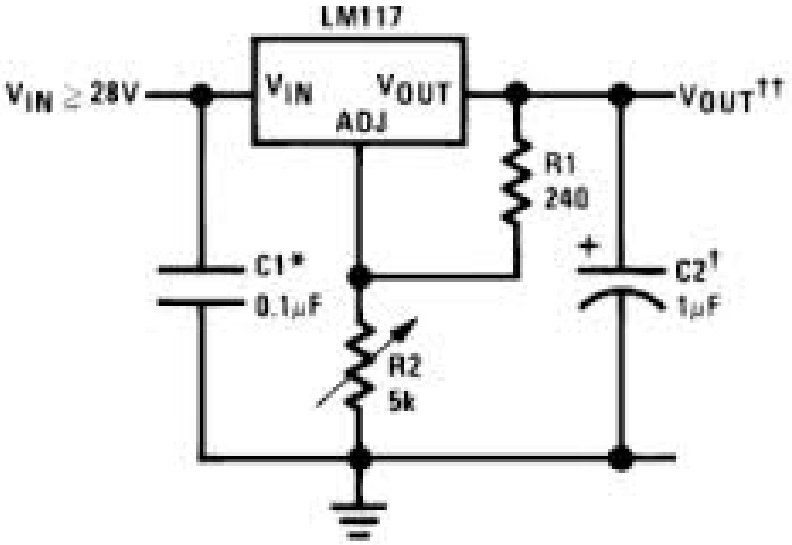
- To power the Launchpad and most other circuitry, you'll want to use a regulated voltage. 3.3 V for the Launchpad, maybe 3.3V or 5V or 15V for other components. (can run MSP430 off of 2 AA or AAA batteries directly).
- These voltages are most easily made with a 3 pin voltage regulator.

- eg LM7805, LM7815, UA78M33
- These can often supply up to 1A, but may need a heatsink



Voltage Regulation

For 'non-standard' voltage, LM317 is a three-terminal, adjustable regulator



The regulator attempts to maintain:
$V_O - V_{ADJ} = 1.25$ V (Vref)

So V_{out} is set by the ratio of R_1/R_2

$V_O = V_{REF} (1 + R_2/R_1) + I_{ADJ} R_2$
$I_{ADJ} = \sim 50$uA.

Choose R_1, R_2 so that I_{ADJ} x R_2 is small, but also so V_O x (R_1+R_2) is not big.
$R_1 = 240\ \Omega$ is recommended.

Power dissipation

- Three pin regulators can get very hot, and may need a heatsink. They tend to draw exactly the same current from the supply as they output, and they dissipate the power difference.

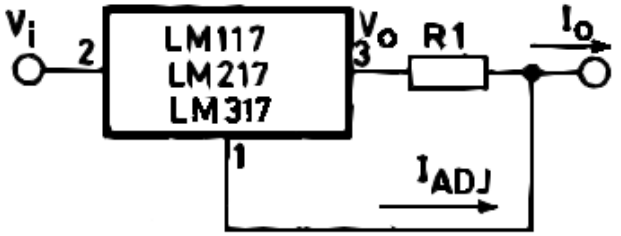
  For example, a 5V regulator operating from a 12V supply, supplying 1A has to dissipate (12V-5V)x1A = 7W. Without a heatsink, this would get very hot, very fast!

Dropout

- Many 3 pin regulators have a fairly high (1.5 – 2 V) “dropout” voltage. This means that for a 5V regulator, the input needs to stay above 6.5-7V.
- There exist “low-dropout” regulators, some of which are also low-power. LP2950 is a nice family.

Current Regulation

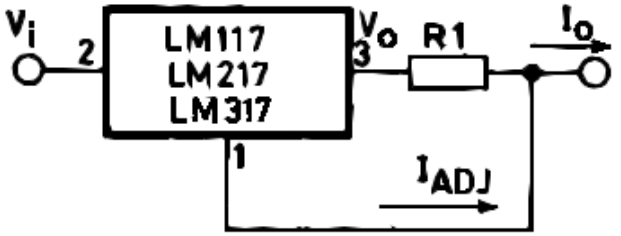
- Most of our projects won't need current regulation, but some motor driving applications may (eg if powered from batteries). A simple way to regulate current is to use the LM317 in a slightly different mode:



$$I_O = (V_{ref}/R_1) + I_{ADJ} = 1.25V/R_1$$

Current Regulation

- Most of our projects won't need current regulation, but some motor driving applications may (eg if powered from batteries). A simple way to regulate current is to use the LM317 in a slightly different mode:



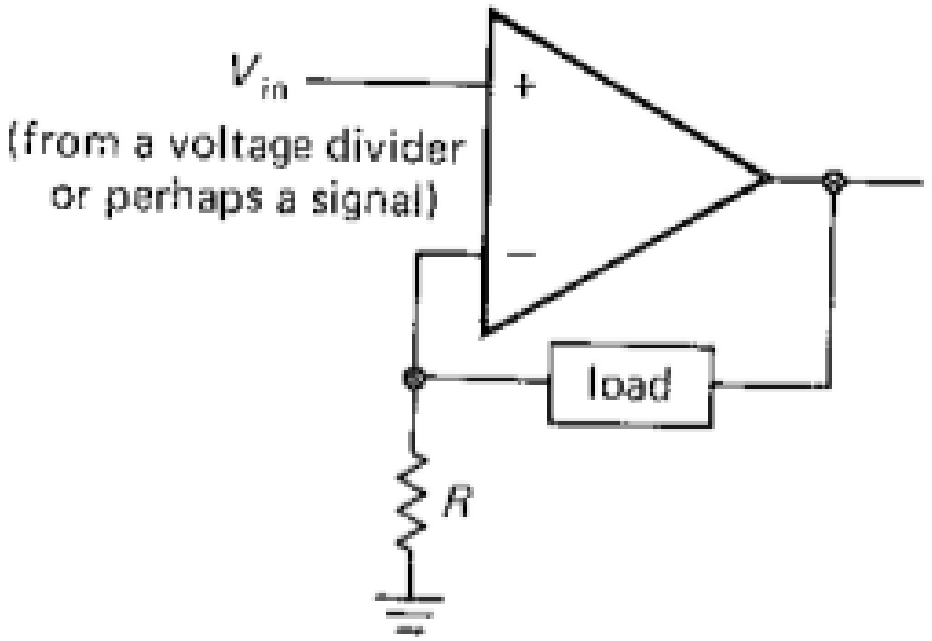
Needs Vin = Vout + ~3V

(1.25 across R1, plus ~ 2V dropout)

A Low-dropout adjustable regulator would help.

$$I_O = (V_{ref}/R_1) + I_{ADJ} = 1.25V/R_1$$

Current Regulation



But load isn't grounded in this circuit.

Current Regulation

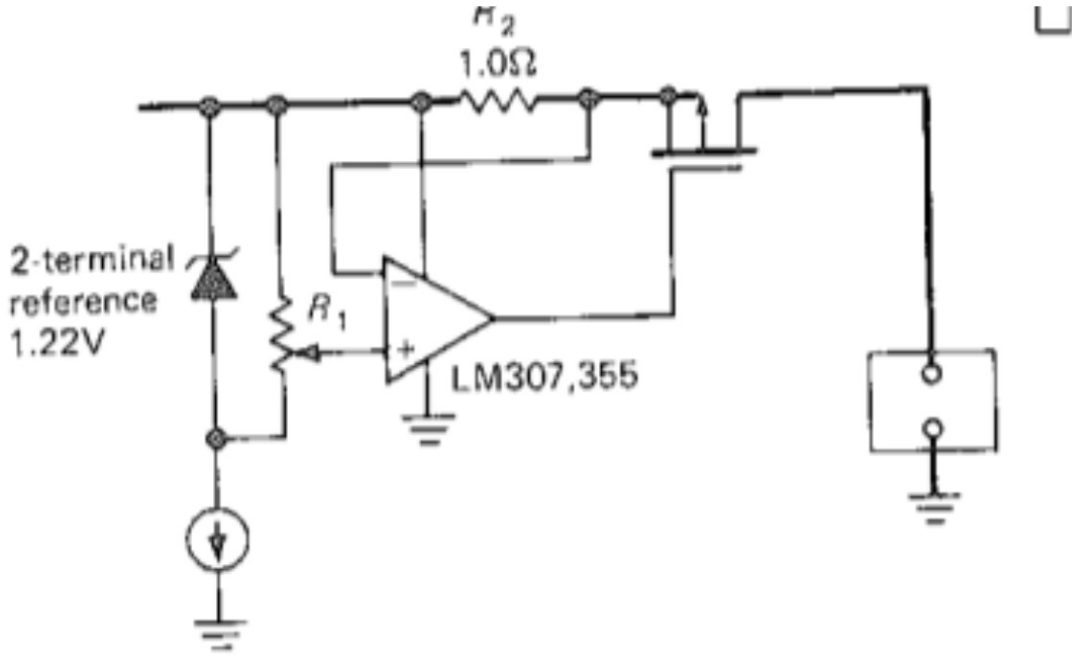
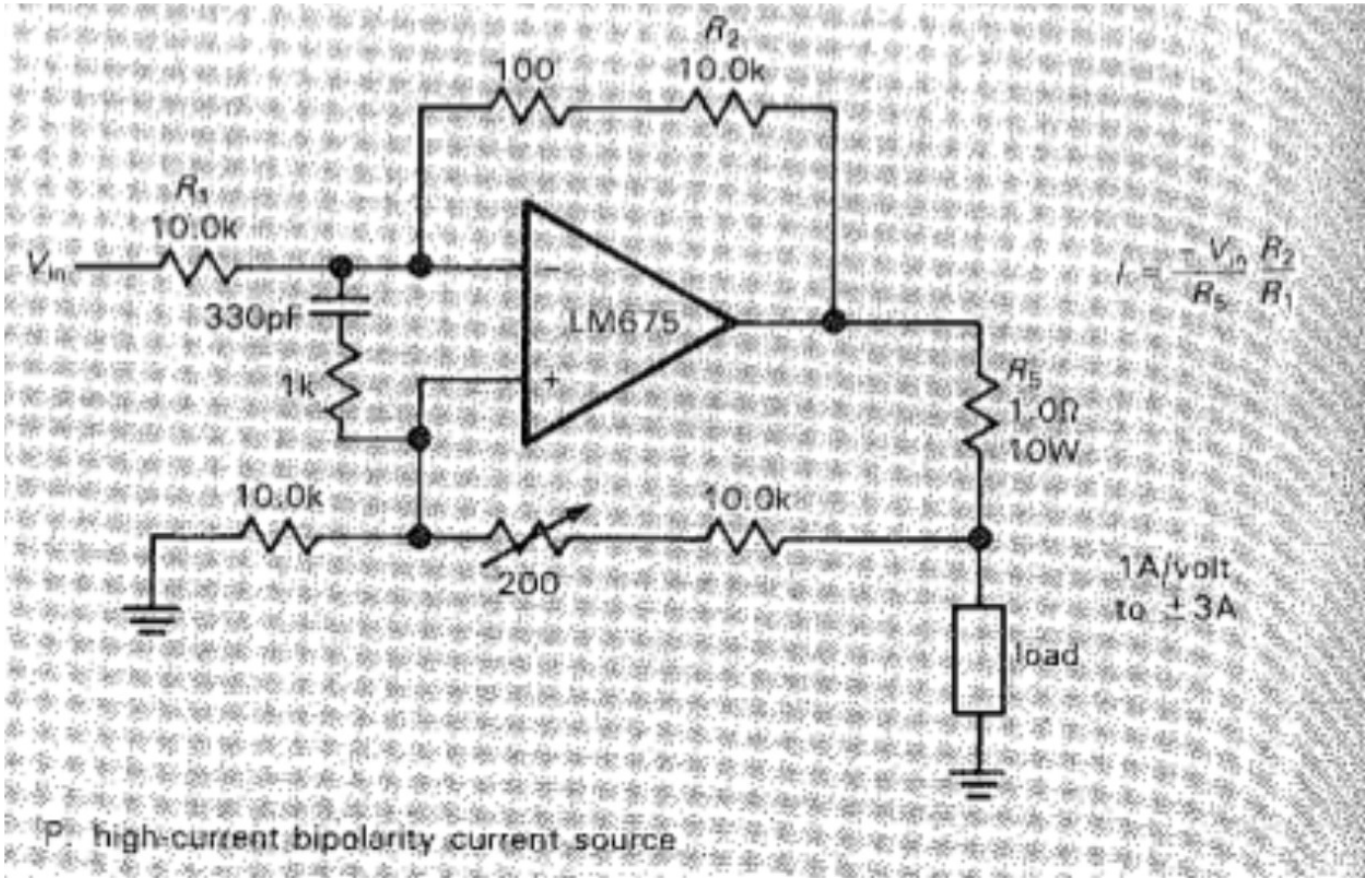


Figure 6.61. Input-rail current sensing.

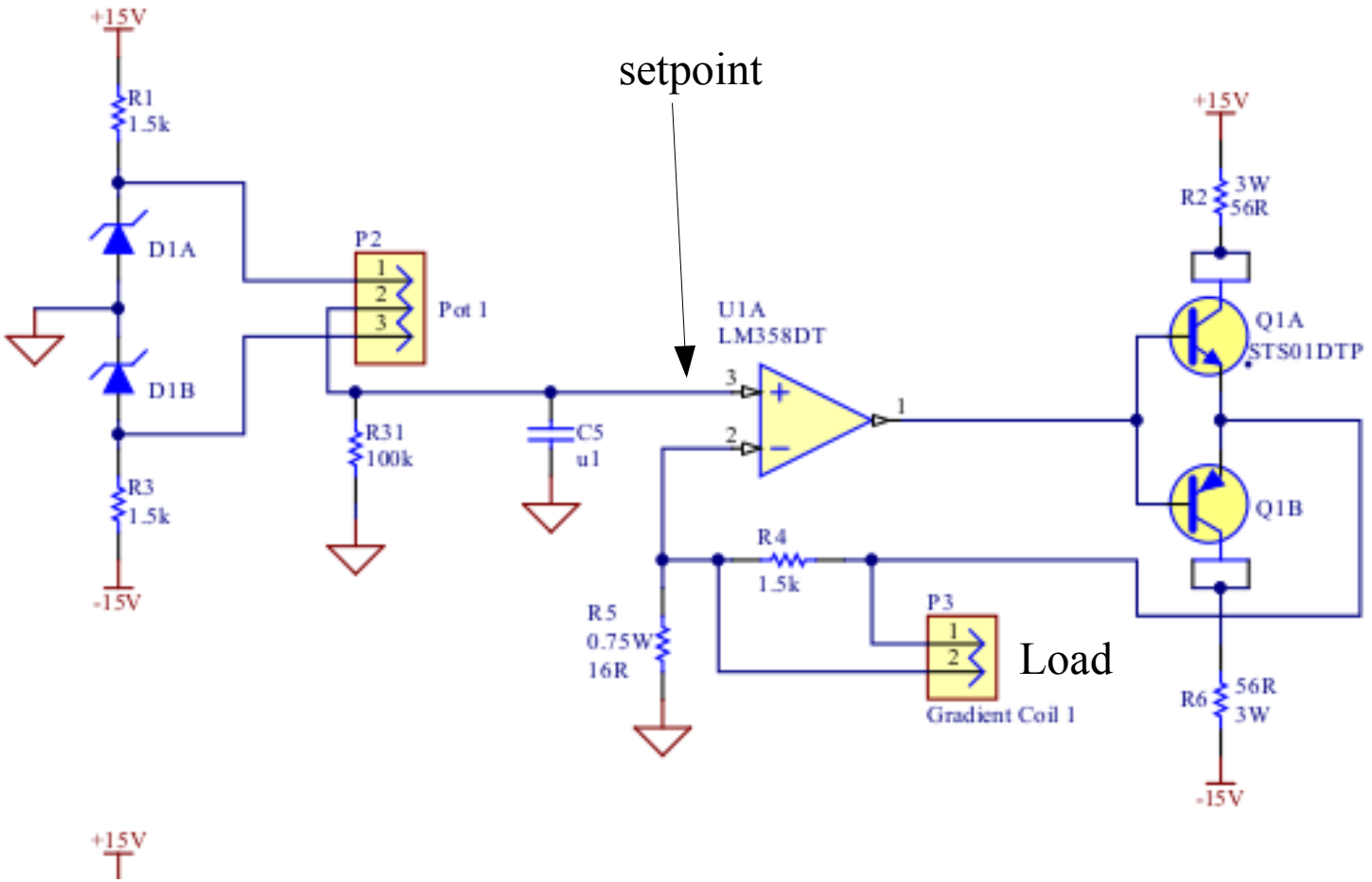
Load is grounded, but drive is unipolar

Current Regulation



P. high-current bipolarity current source

Current Regulation



Current Regulation

For a low impedance motor winding, something based on this circuit might work?

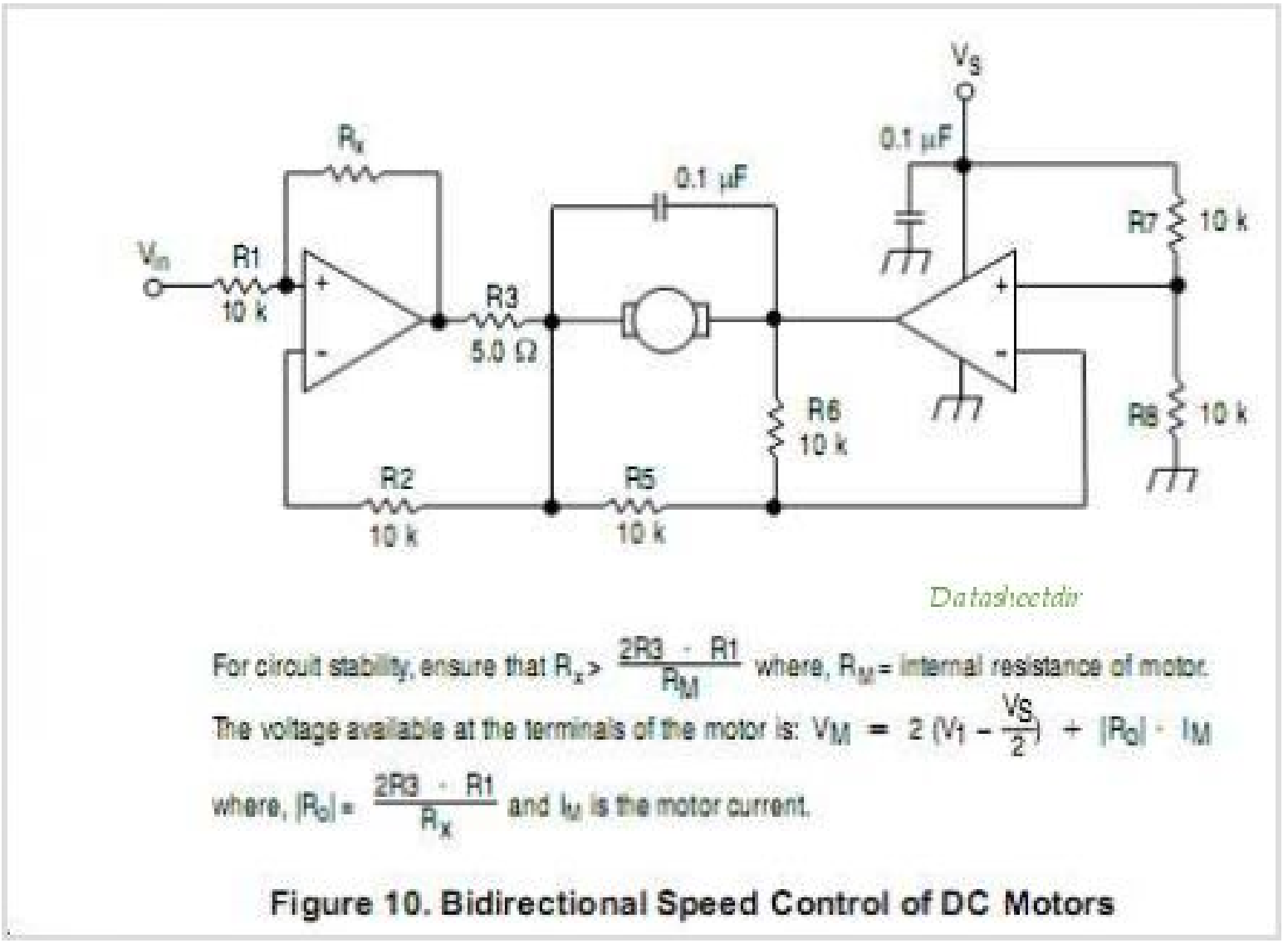
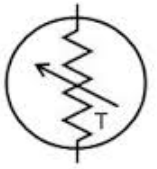


Figure 10. Bidirectional Speed Control of DC Motors

Overcurrent Protection

- If using a DC power supply, generally it should be chosen so that it can supply the needed amount of current, and not too much more.
- For battery powered projects though, it may make sense to include overcurrent protection to avoid: damaging the batteries or starting a fire.
- Options include: fuses (kind of a pain as it needs to be replaced), circuit breakers (expensive), thermal cut-outs (cheap, basically a fuse), PTC thermistors, or for a low power project, just a resistor in series with the supply may be ok (must be capable of dissipating the power developed in it when the load is short circuited).
- The PTC thermistor is a semiconductor device where the resistance increases rapidly with temperature. If you try to draw “too much” current, the resistance rises and reduces the current.



DC-DC convertors

- step-up or step-down DC-DC convertors are available.
- (Much) more energy efficient than linear regulators
- Noisier output
- Either expensive or requires more supporting components (an inductor !)

  (becoming less true)
- Allows to generate 5V from 2 AA batteries.