Signal Processing in Software and Electric Field Sensing

CSE 466: Software for Embedded Systems
Winter 2009

B. Mayton <bmayton@cs.washington.edu>
University of Washington CSE & Intel Research Seattle

Labs 3 and 4: Building a Sensor

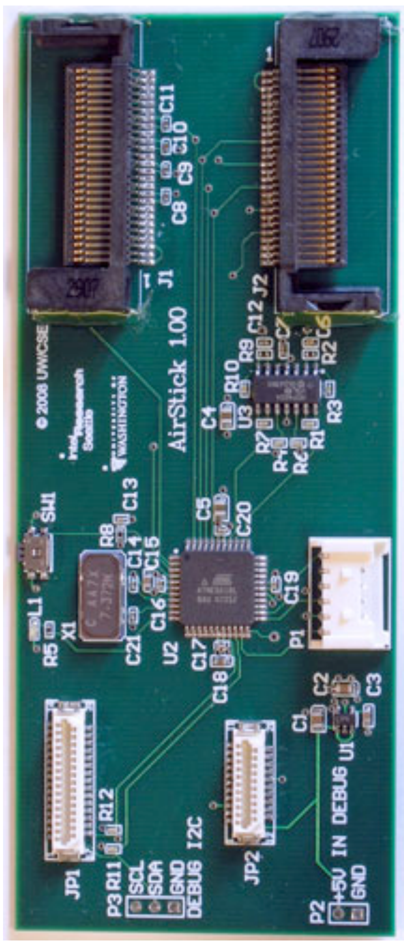
- You'll build an electric field sensor with basic components and your AVR microcontroller
  - Can sense your hand above the board without actually touching it
  - Transmit and receive signals are generated and demodulated in software
  - Will send the sensor readings to a PC for further processing and moving around in a color space

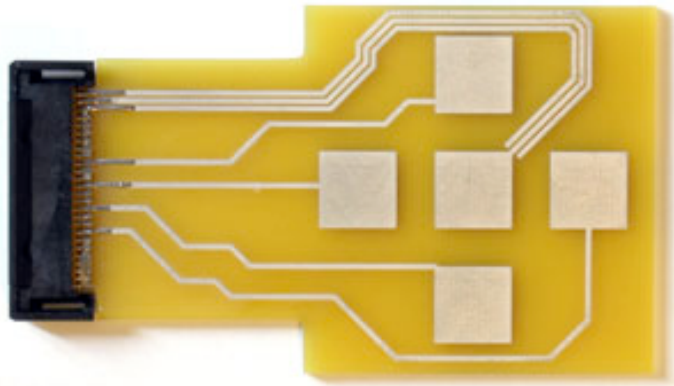
E-Field Sensing in Nature

Black Ghost Knifefish
(*Apteronotus albifrons*)

- Generates an electric field around 1 kHz
- Uses E-field to detect prey, avoid objects, and communicate

Electric Field Sensors: Applications

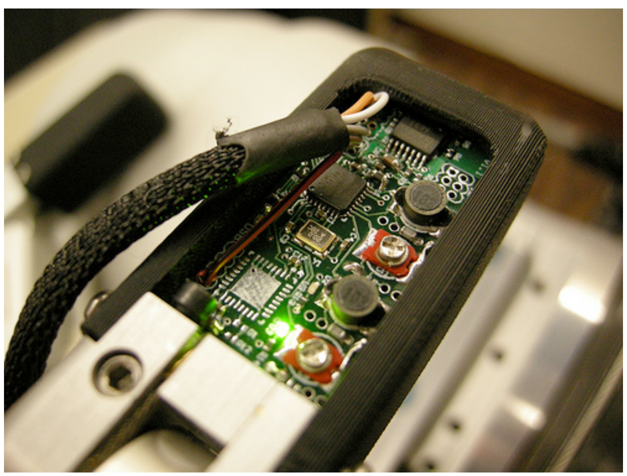




- CSE 466 Winter 2008
  - added 4-channel E-field sensor (the "AirStick") to the iMote 2 to allow control of a soccer player in a multiplayer game

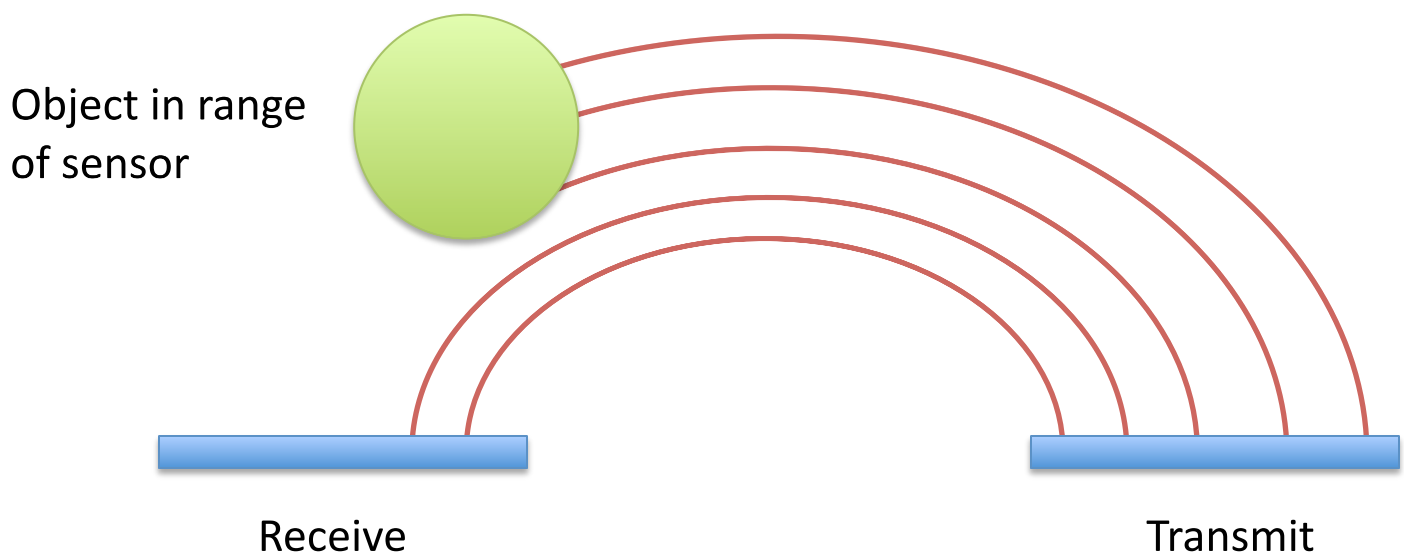
Electric Field Sensors: Applications

- Personal Robotics at Intel Research Seattle
  - E-Field adds sense of "Pretouch" to a robot hand and arm, allowing it to detect that it is close to objects without touching them



Theory of Operation

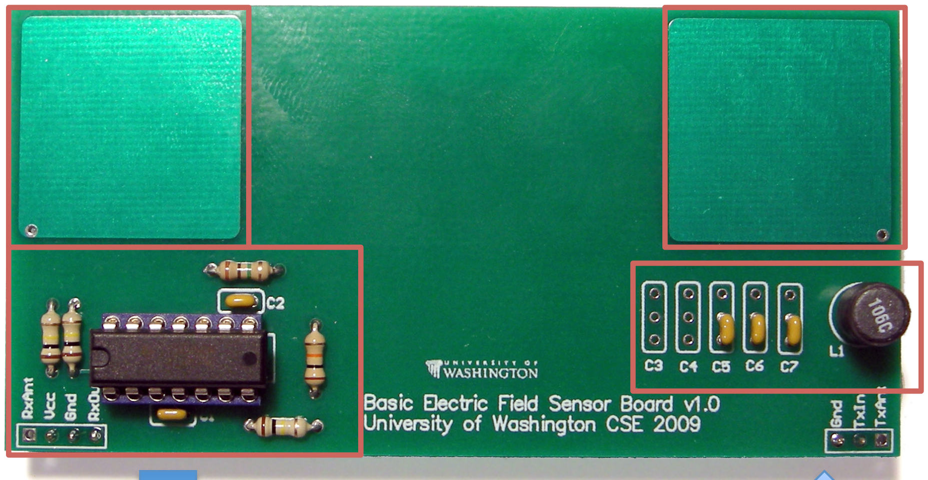
- Create an AC electric field with constant amplitude with transmit electrode
- Measure current induced in receive electrode
- Nearby grounded objects shunt some current to ground, reducing the received current.



Parts of the Sensor

Receive Antenna

Analog Front-End



Transmit Antenna

Transmit Resonator

Sampling (ADC)

Demodulation (Software)

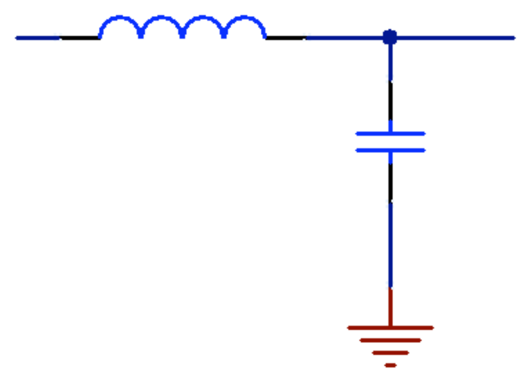
Waveform Generation
(Timer 0 Output Compare)

Transmitting an Electric Field

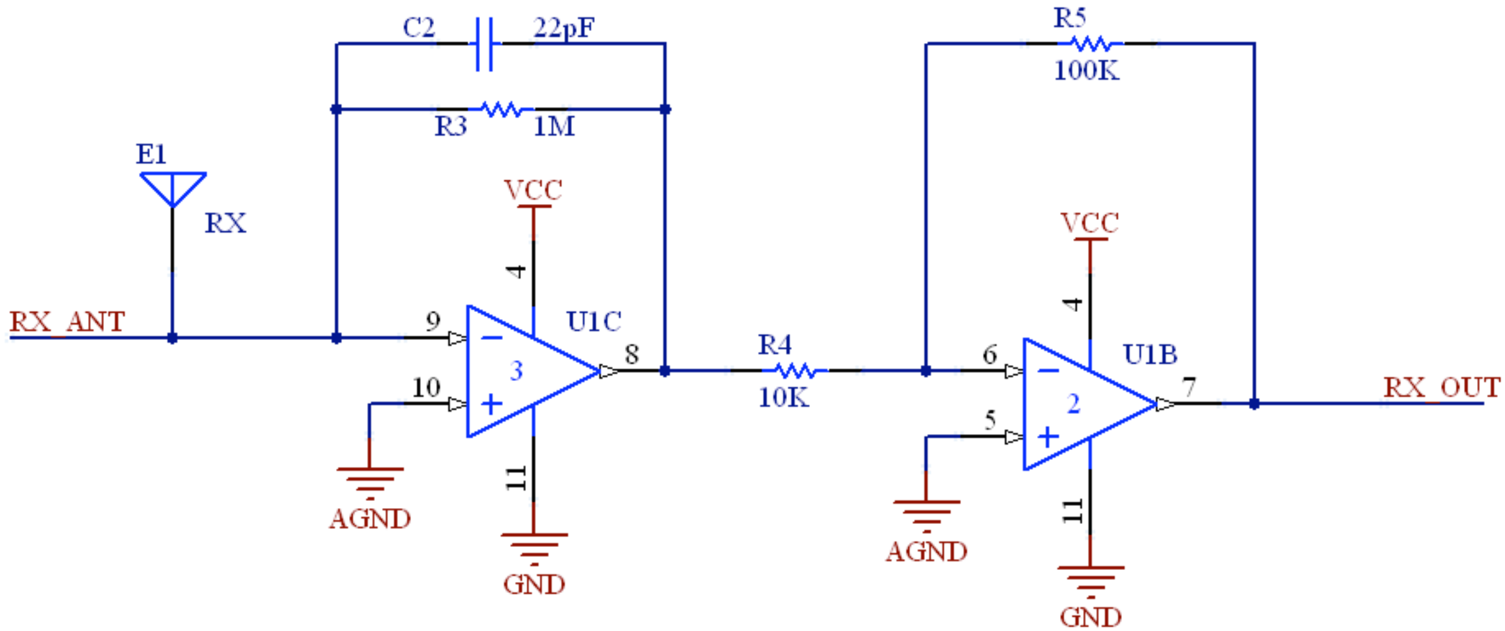
- Use timer hardware on AVR to generate a square wave; frequency can be tuned with prescaler value and output compare value
- Resonator circuit boosts 5V square wave to a sine wave with an amplitude of around 100V
- Transmit in the tens of kHz range; higher frequencies work better but give less time to process interrupts

Resonant Circuits

- Driving a resonant circuit with small amounts of energy at the right times (at the resonant frequency) will cause high-amplitude oscillations
  - The LC resonator on the sensor board allows the AVR to create a large electric field from a supply of only 5V
  - The resonant frequency of an LC circuit is $f = \frac{1}{2\pi\sqrt{LC}}$

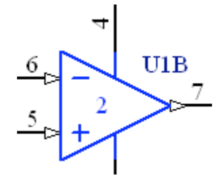


Amplifying the Received Signal



- Transimpedance amplifier converts current flowing into received electrode into voltage
- Voltage gain stage amplifies the signal to levels that work well with the ADC in the AVR

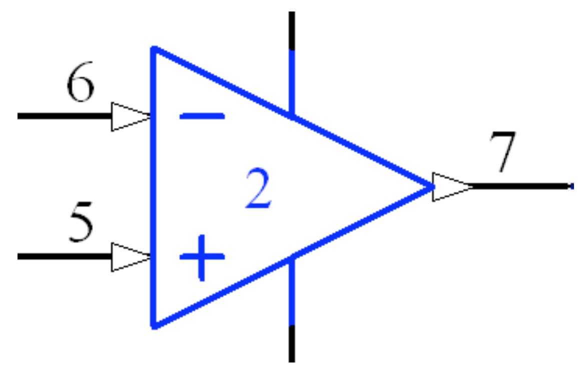
Operational Amplifiers



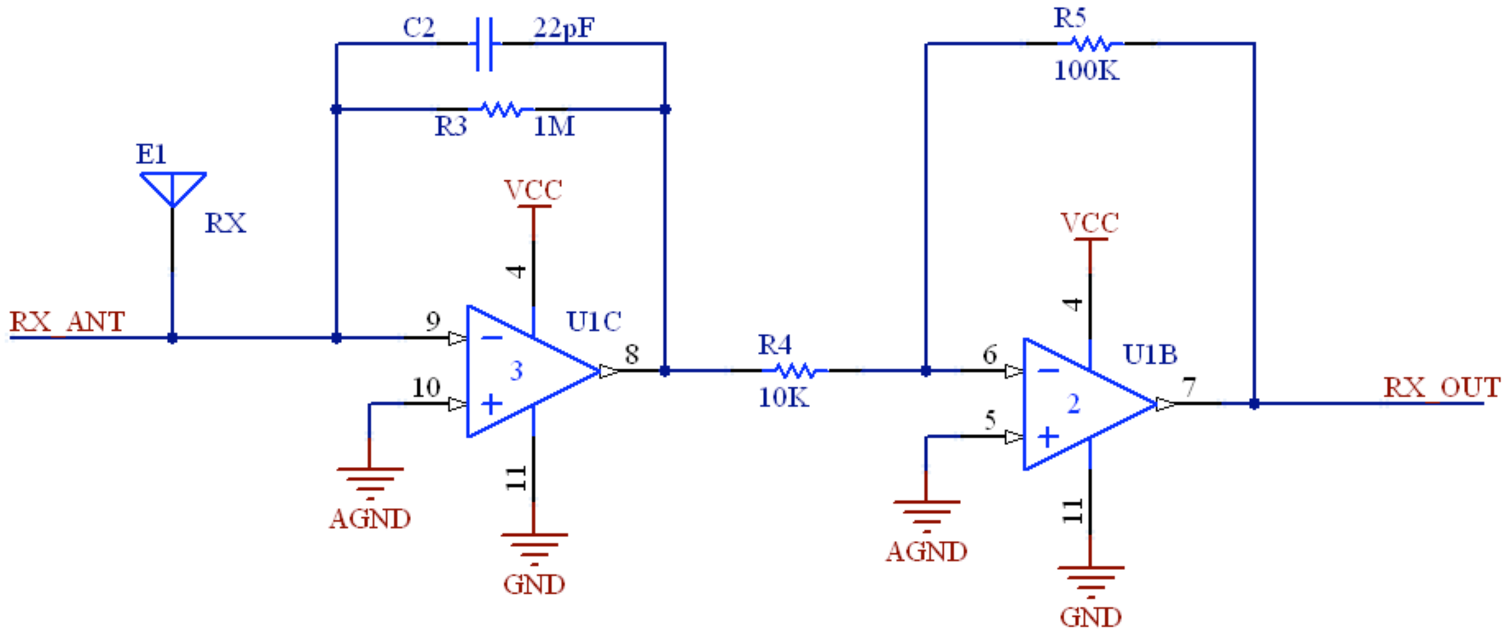
- Very useful, versatile, and ubiquitous analog circuit devices
  - amplify voltages
  - can turn high-impedance signals into low-impedance signals (weak signals into robust signals)
  - perform mathematical operations on signals in the analog domain (used to be how most signal processing was done)
- Signal processing has moved into the digital domain, but op amps are still useful, particularly when it comes to interfacing sensors with microcontrollers

Operational Amplifiers

- Two input terminals: inverting (-) and non-inverting (+)
  - almost no current flows into the inputs (they are *high impedance*)
- Voltage at output terminal is the difference between the two inputs multiplied by some gain
- Output changes to try to keep the voltages at both inputs equal to each other
- Output is *low impedance*: we can draw some current from it without affecting its voltage significantly



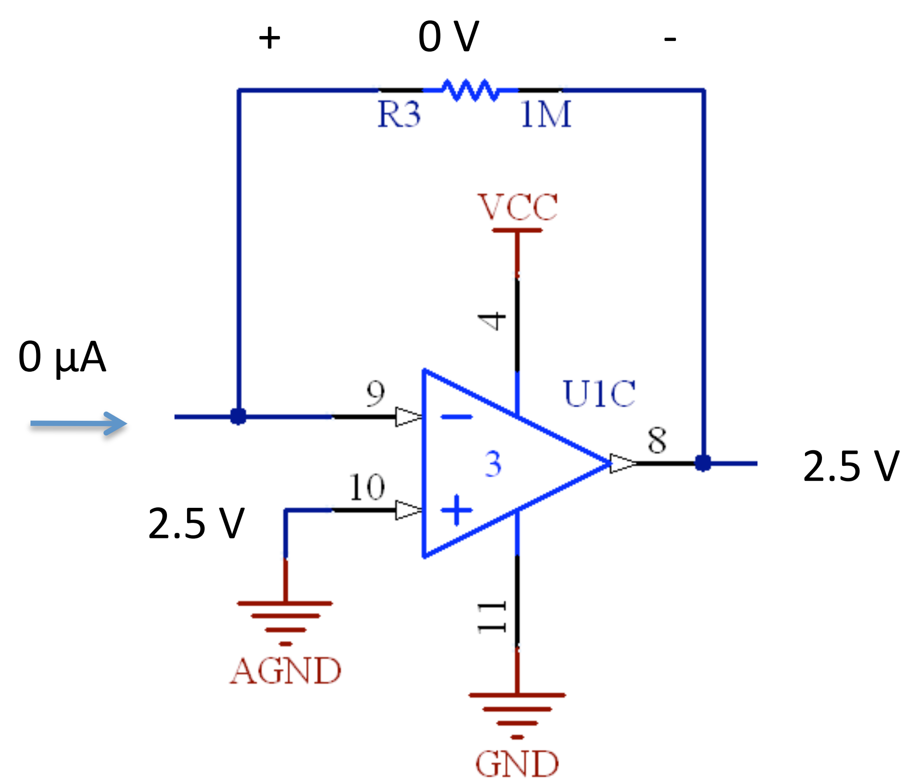
The Analog Front-End



- Now that we know about op amps, we'll look at the two stages of the analog front-end.
- AGND is at 2.5V; this will add a DC offset to the output voltage, bringing "zero" into the middle of our usable range for the ADC

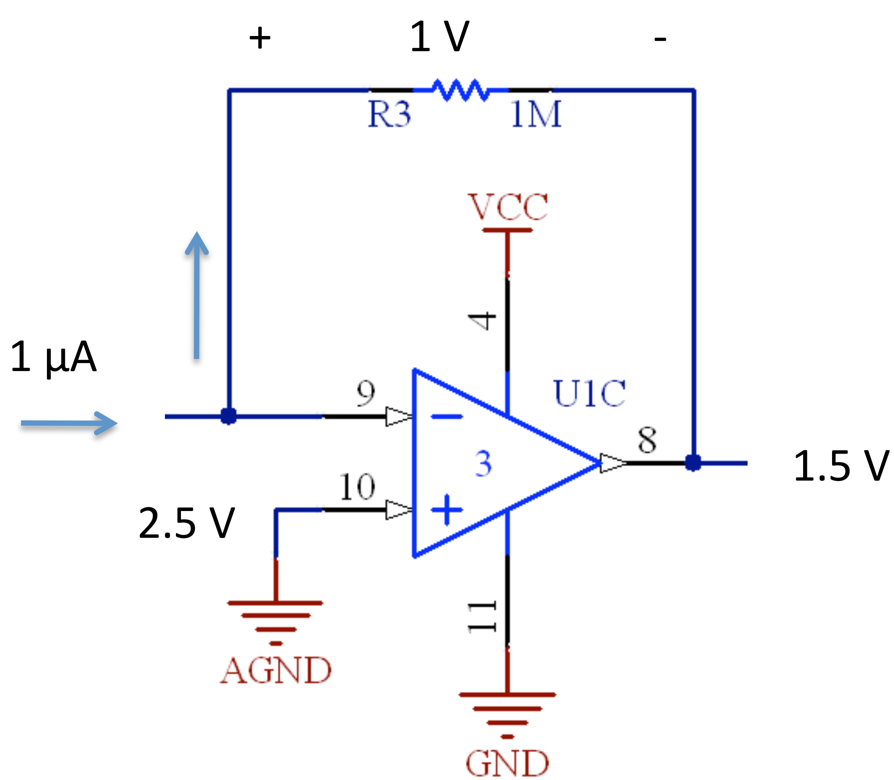
The Transimpedance Amplifier

- Converts current entering the receive electrode into a voltage signal
  - When no current is flowing in from the left, there is no voltage drop across the resistor, and the output voltage will be the same as the two input terminals.



The Transimpedance Amplifier

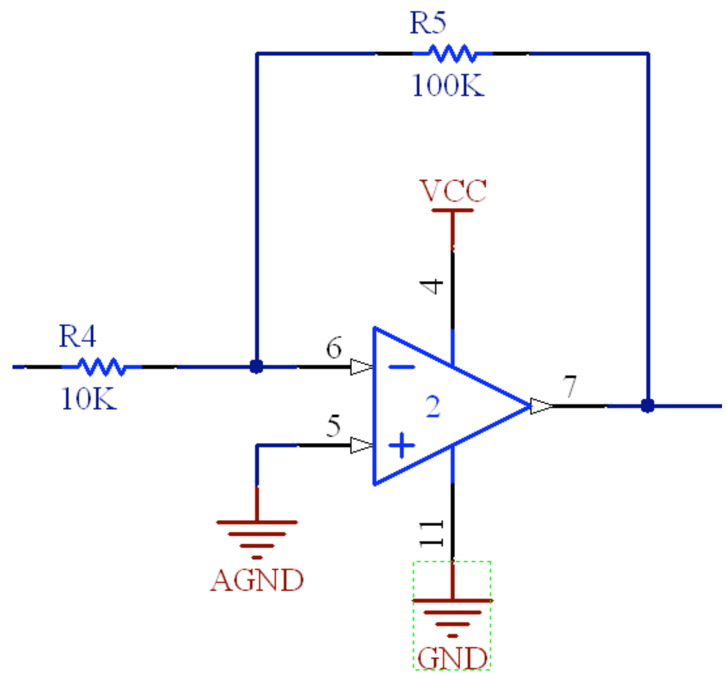
- Converts current entering the receive electrode into a voltage signal
  - Current entering from the left can't go into the inverting input, so it goes through R3 and creates a voltage drop
  - In order to keep the voltage at the inverting input equal to the non-inverting input, the output voltage must be decreased



The Voltage Amplifier

- Amplifies (and inverts) the input voltage.
- Basic inverting op-amp configuration
- Output voltage for the circuit at the right is

$$v_{out} = -v_{in}\frac{R5}{R4}$$



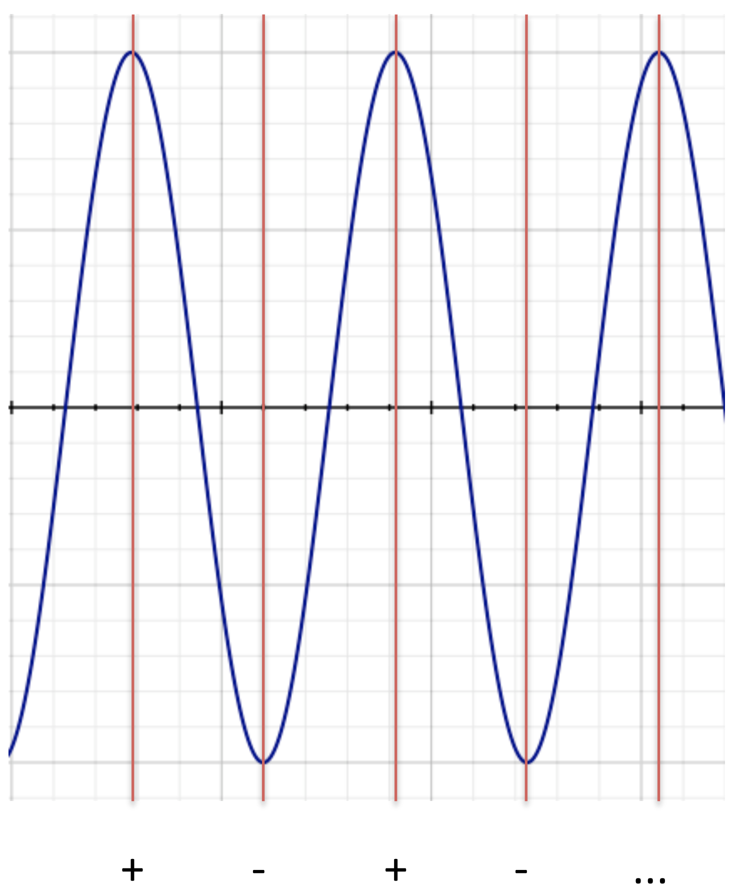
Where are we now?

- We’ve created an electric field and it’s induced a current in our receive electrode.
- We’ve amplified the received signal and brought it into a usable range for our AVR’s ADC to sample.
- Now, we need to make sense of the signal

Demodulation

- We'll be receiving our signal, but there will also be a lot of noise.
- Need to recover the amplitude of our signal, but ignore the noise.

Demodulation: Basic Idea



- Instead of one sample, we'll *accumulate* multiple samples in an intelligent manner.
- Sample at multiple points on the received waveform
- If we add samples when we're transmitting a positive signal and subtract signals when we're transmitting negative signal, we cancel out a lot of unwanted noise (and the DC offset)
- Accumulate about 20 to 255 samples for a measurement
- Result of accumulation represents amplitude

What about phase shifts?

- If the samples aren't perfectly in phase with the received waveform, we're missing out on signal-to-noise
- Getting perfectly lined up is difficult—at high transmit frequencies, even one instruction cycle shifts the sample by quite a bit
- Solution: also sample at 90˚ and 270˚ in addition to 0˚ and 180˚, in a separate accumulator
- The accumulator for 0˚ and 180˚ is the in-phase component, and the 90˚ and 270˚ accumulator is the quadrature component
- We can now recover the magnitude of the received signal regardless of its phase:

$$magnitude = \sqrt{inphase^2 + quad^2}$$

Timing Issues

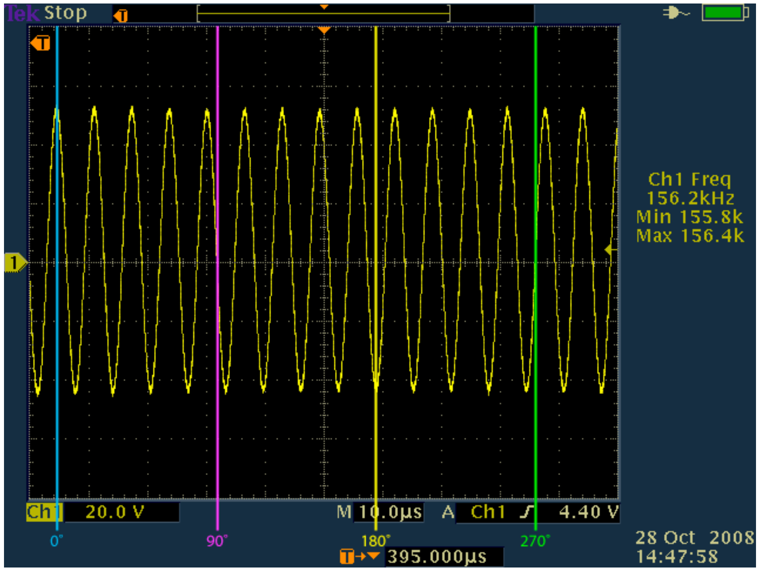
- The ADC isn't fast enough to make all these samples on every period of the received waveform.
- Lining up the samples with the right parts of the waveform is a challenge.

but…

- The ADC takes some time to perform conversions, but the actual sample-and-hold window is short and can be precisely placed.
- We can make the ADC's samples synchronous with the transmitted waveform by using free-running mode and setting up the prescalers so that the ADC samples every n + ¼ periods of the transmitted waveform.

Timing

- The result is that we're only sampling every few periods of the waveform, but we stay lined up because we're using the same clock for the transmitted signal and the ADC.
- Since we set up the ADC to sample every n + ¼ periods of the transmitted waveform, each sample will be offset by 90˚.
- Each set of four samples gives us the positive and negative values to add to each of our two accumulators.



Implementation in Labs 3 and 4

- For labs 3 and 4, you will:
  - calculate prescaler and output compare values to enable synchronous undersampling and demodulation
  - calculate the capacitor value needed to make the transmitter resonant at your transmit frequency
  - build the e-field sensor hardware
  - use SPI and USB to send the values of the in-phase and quadrature accumulators to a PC
  - use the PC to compute the magnitude of the received waveform from its components
  - use the value from the sensor as a virtual knob to move around in a color space
  - display a color wheel indicating the current color on the PC screen, and send it back to your AVR to be displayed on your tri-color LED