IOActive™

COMPREHENSIVE COMPUTER SECURITY SERVICES

RFID for Beginners

Chris Paget, Director of R&D, IOActive

chris.paget@ioactive.com

About this presentation

- Basic electronics knowledge is assumed
  - Voltage, current, resistance, etc
- You won't need more than the basics
  - Don't need to know about radio
  - Don't need to know about RF design
- Everything will be explained…
  - Inductance
  - Capacitance
  - Etc…

About RFID

- Radio Frequency IDentification
- Many different kinds of RFID
  - Active Vs. Passive
  - ID-only Vs. Encrypted
- Many different uses for RFID
  - More being found all the time
- Tags can be found in some surprising places…

Active Vs. Passive

- Active tags:
  - Have their own power source
  - Are low-power radio transmitters
  - Medium-distance (several meters)
  - Small devices
    - Smaller than a deck of cards is typical
- Listen for a signal from the reader, and broadcast a response
- True electromagnetic radio transmitters

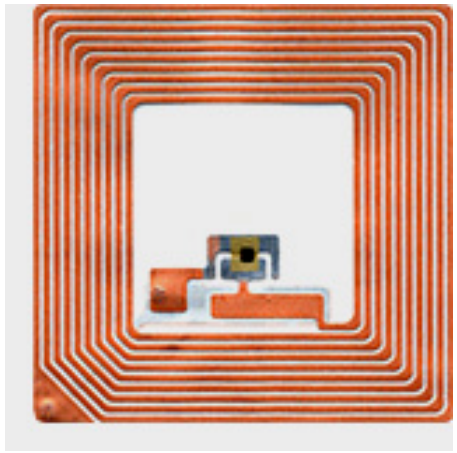
Active Vs. Passive

- Passive tags:
  - Powered parasitically by the reader
  - Use Inductive Coupling, not radio
  - Very short range (a few inches)
  - Very small indeed
    - The size of a grain of rice, including antenna
    - Usually the coil is a few inches to a side
- Used in different situations from active tags
  - Small size considerations
  - Read range is not a factor

Active Vs. Passive

- Active tags:

- Passive tags



ID-only Vs Encrypted

- ID-only tags simply send the same code every time
  - Easy to clone
- Encrypted tags handshake with the reader
  - More difficult to clone
  - Many “encrypted” tags can be broken
- Active or passive tags can use either method
- ID-only is far more common

Uses for RFID

- Toll roads
  - "Speedpass" express lanes
  - Active tag, read at high speed
- Verichip
  - Implantable, passive tag
  - Used in animals for some time
  - Used in humans more and more
- Access badges
  - Normally ID-only, passive tags

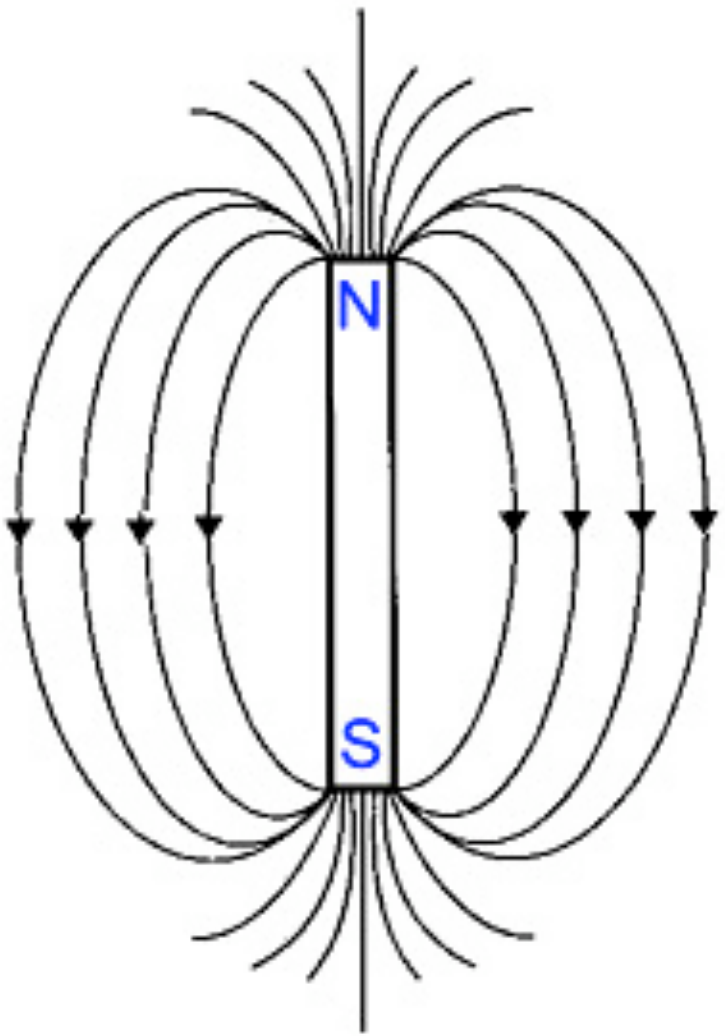
Mechanism of Operation

- HID access cards
  - Passive tag
    - Inductively coupled with the reader
  - ID-only
  - 125KHz carrier frequency
    - VERY simple to mess about with
  - Simple protocol
    - Preamble
    - 506-bit data sequence
    - Postamble

Magnetic Fields

- Attract ferrous materials
- Decrease with distance
  - Inverse Cube law
    - Fundamental reason why passive tags are low-range!
- North and South pole
  - A magnetic field has direction

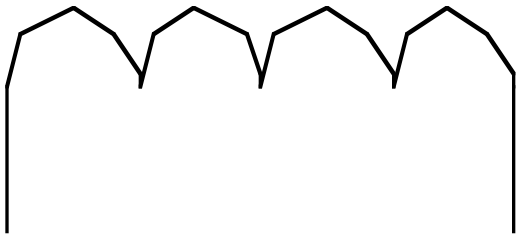
A Magnetic Field

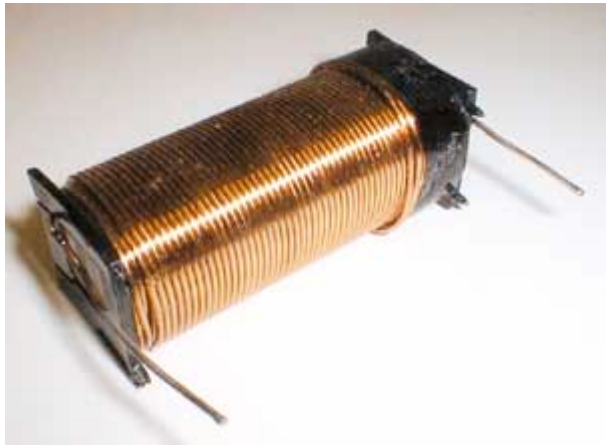


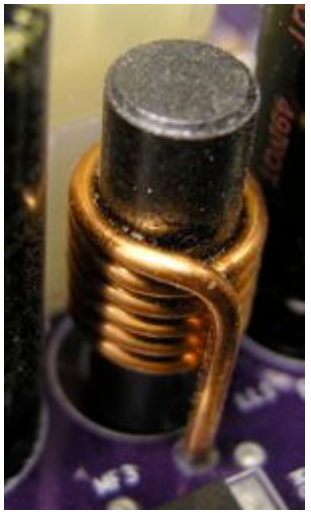
Induction

- A changing current produces a magnetic field
- Field strength increases with rate of change of current
  - Superconducting magnets use 10,000A+
- Coils of wire build the field strength
  - More loops, stronger field

Inductors

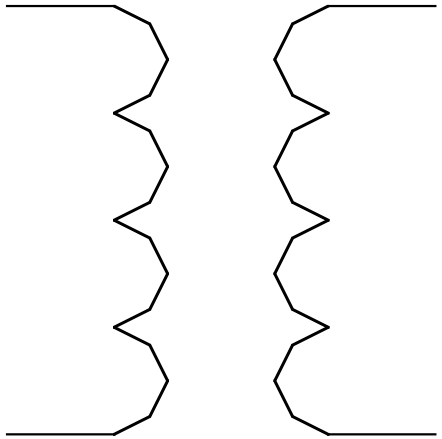






Inductive Coupling

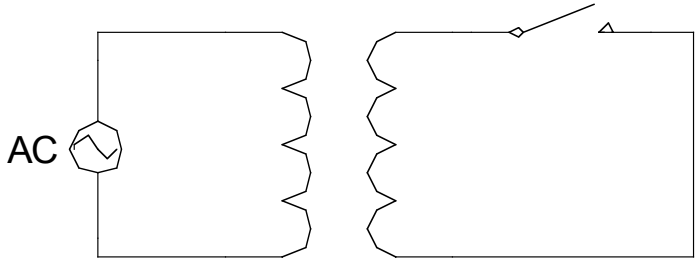
- A changing magnetic field causes a current to flow
  - Induction works both ways!
- Two inductors = a transformer



Inductive Coupling

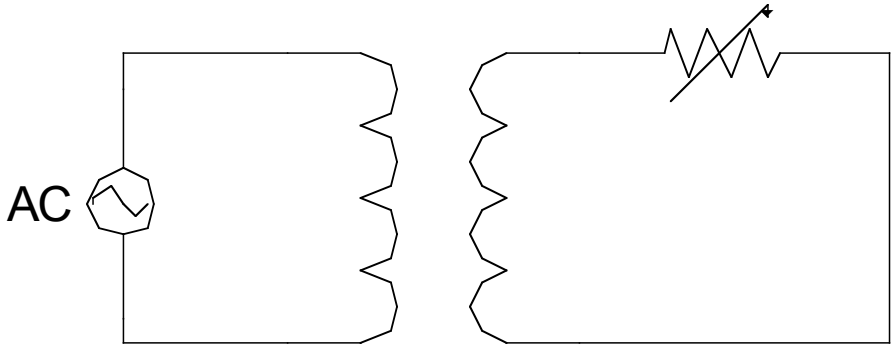
- Apply a changing current to one coil
  - The Primary
- This causes current to flow in the other coil
  - The Secondary
- Current flowing in the secondary induces its own magnetic field
  - Opposite direction to the primary
- Counteracts the current in the primary

Signal Transmission



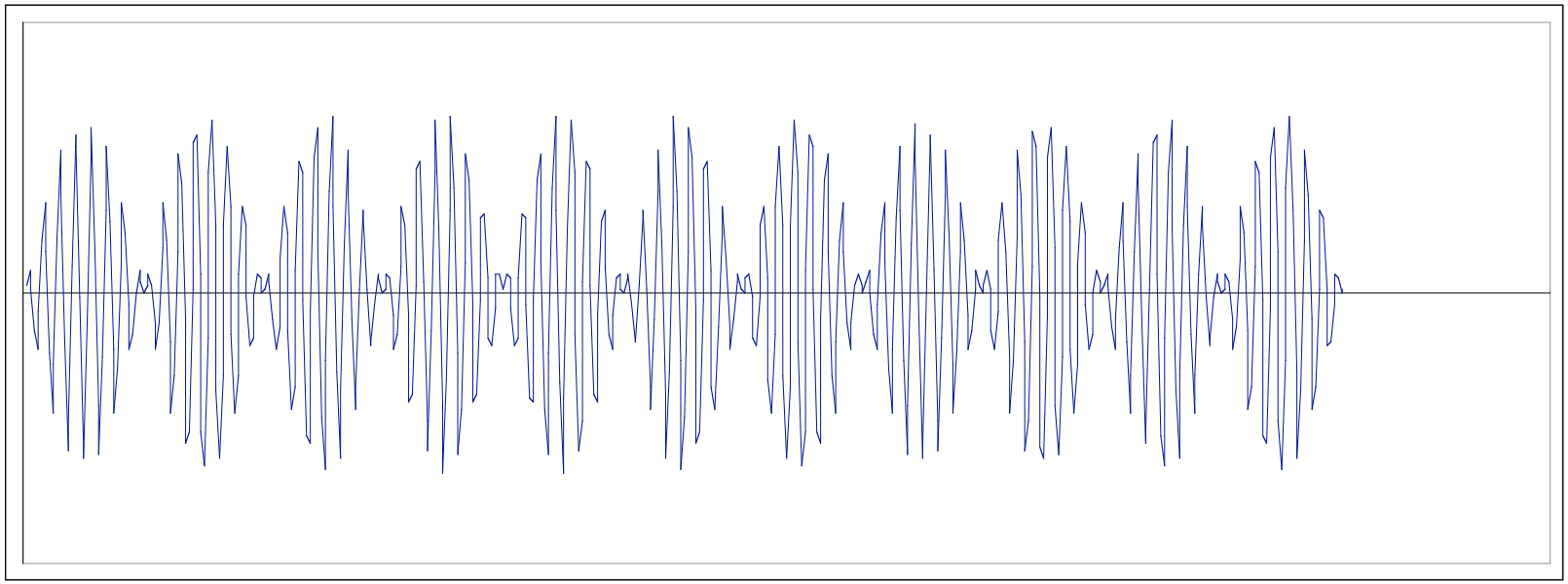
- Switch open:
  - No current in the secondary
  - No opposing magnetic field
  - No change in primary current
- Switch closed
  - Current in the secondary
  - Opposing magnetic field
  - Change in primary current
- Send data by opening and closing the switch

Amplitude Modulation



- Vary the resistor, control how much current in the secondary
  - Controls the strength of the secondary magnetic field
  - Affects the strength of the primary current

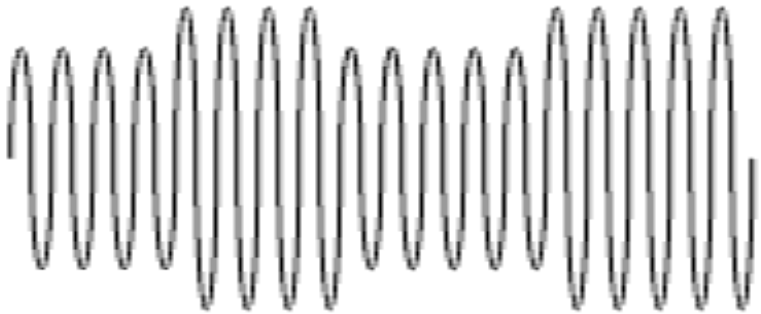
Amplitude Modulation



- Two frequencies:
  - AC frequency (Carrier)
  - Modulating frequency from resistor (Data)

Modulation Schemes

- AM transmits analogue data
  - Need some way to represent binary
- FC/8/10:



- 8 carrier waves for 1 data wave: Binary 0
- 10 carrier waves for 1 data wave: Binary 1

Receiving the signal

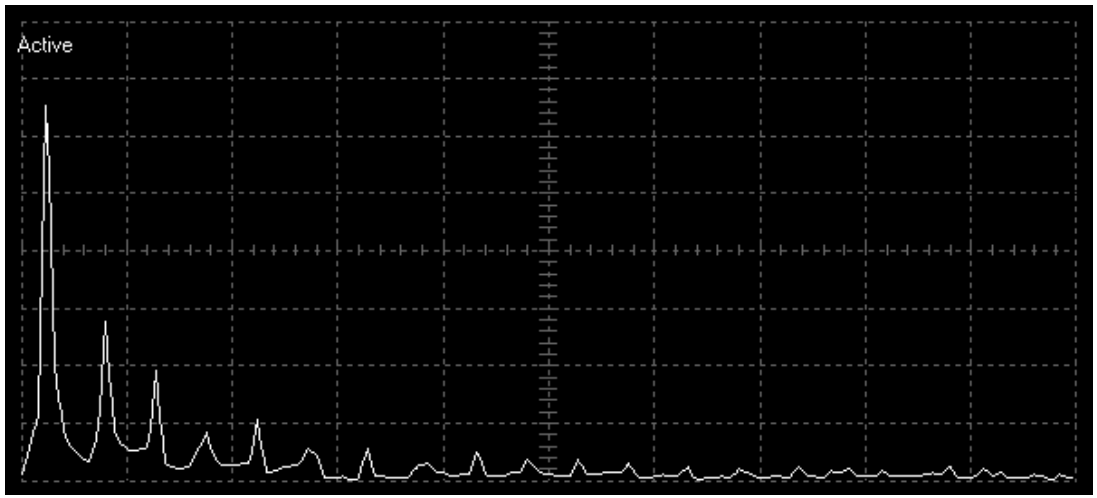
- We know how the data is transmitted
- Need to decode it!
  - Currents through coils of wire aren't much use
- Generate a carrier
- Demodulate the data
- Parse the datastream

Generating the Carrier

- Magnetic field strength is from the rate of change of current, not the current itself
- We want to maximize the field strength
  - Maximize the change in current
- Use a square wave

Square wave noise

- Any waveform can be broken into sinewaves
  - Fourier Analysis
- Square wave frequency components:



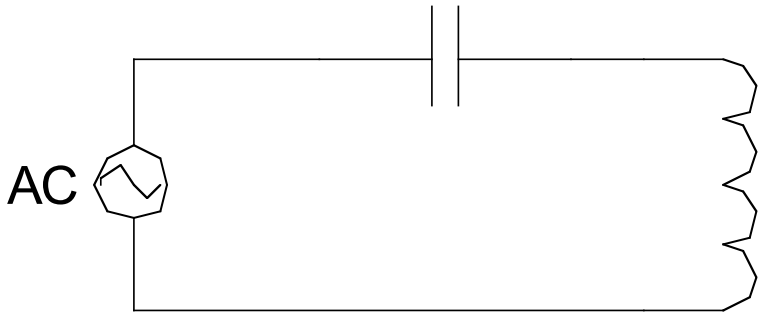
- Too much noise!

Tuning the Circuit

- An inductor resists changes in current
  - By inducing a voltage change
- A Capacitor resists changes in Voltage
  - By inducing a current
- Both are derived from rates of change
- At some point, they'll coincide

$$f = \frac{1}{2\pi\sqrt{LC}}$$

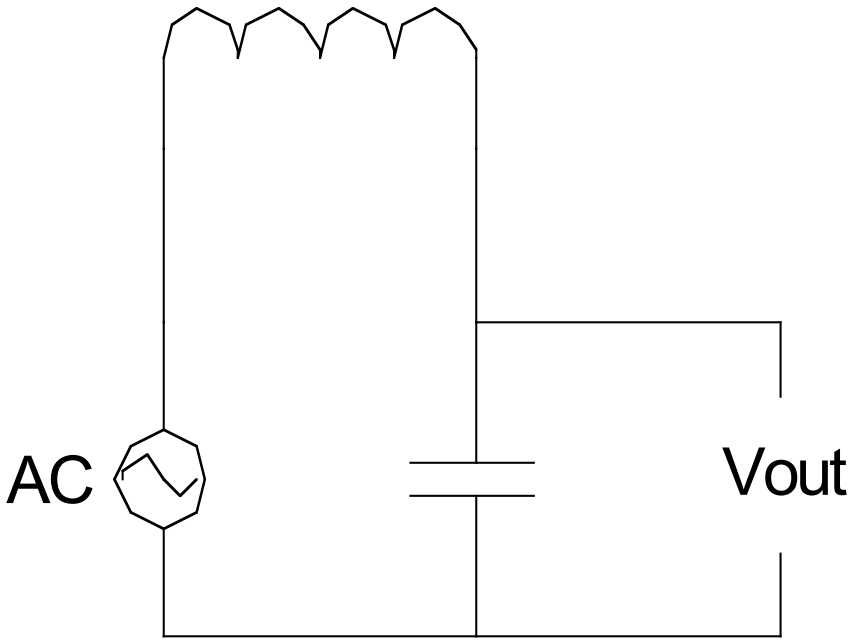
A Tuned Circuit



- AC input is a square wave
- Resonant frequency gets selected
  - And greatly amplified
- All other frequencies get rejected
  - Still present, but greatly reduced
- Sharp voltage changes - strong magnetic field
- Low noise

Receiving the carrier

- Capacitor induces current
- Inductor induces voltage
- Measure the voltage across the capacitor

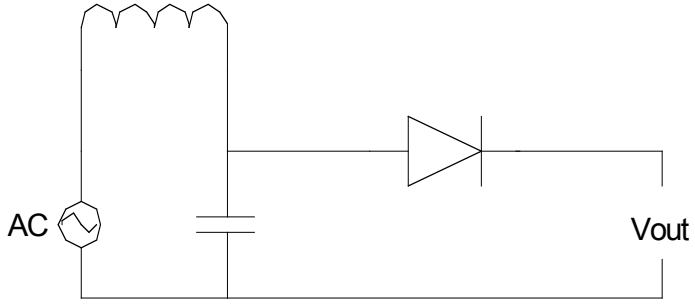


Demodulating

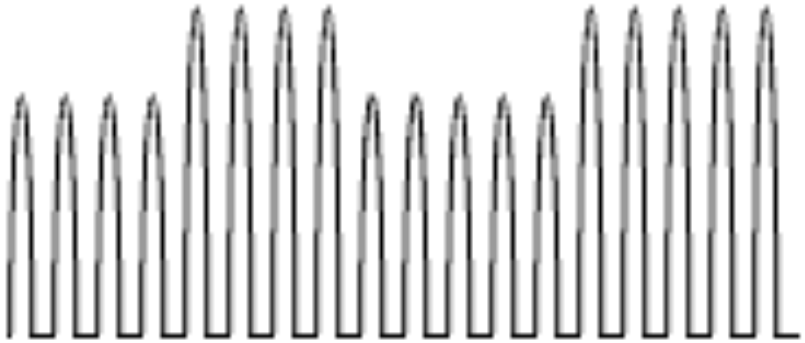
- Remove the carrier and leave the signal
- So far we've been maximizing the carrier
- Problems of scale:
  - 200V of carrier (from a 5V power supply)
  - 2V of modulation (at close range)
- Carrier goes positive and negative
  - Averages out to zero
- First step: Remove the negative half

Rectifying the signal

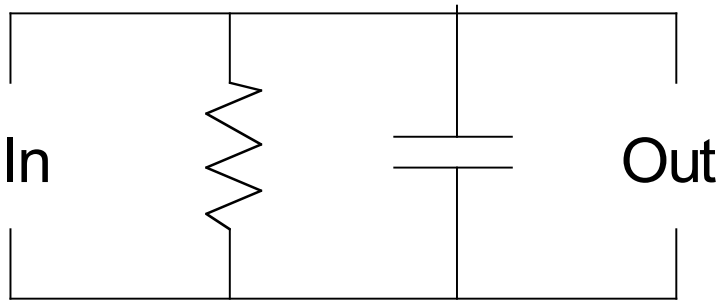
- Diodes only conduct current one-way



- Result:



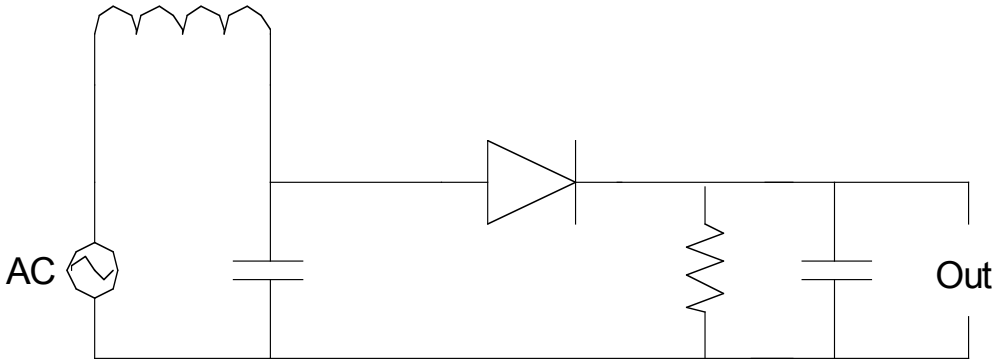
Peak Detection



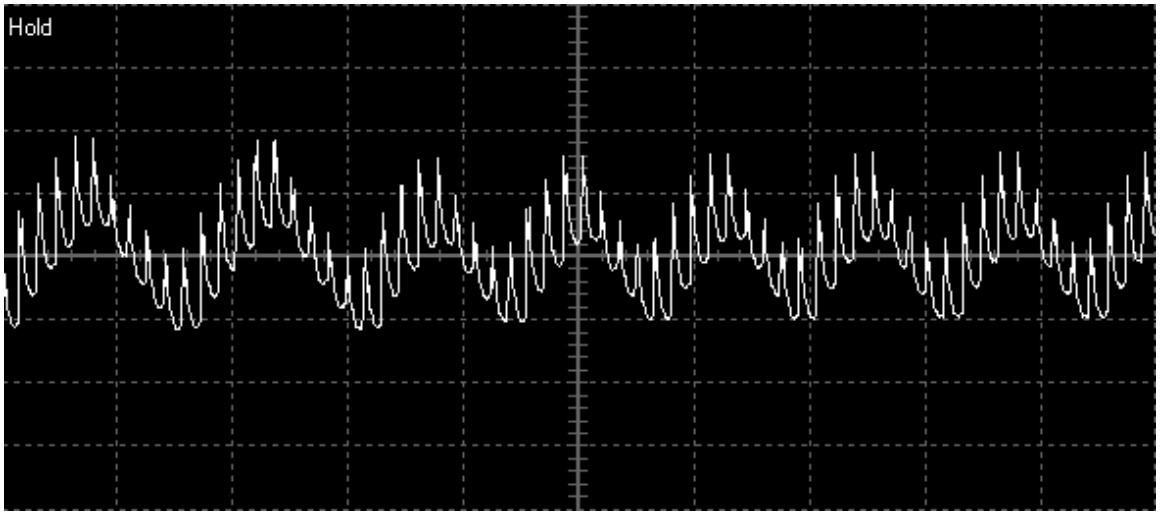
- Capacitor charges when the carrier is high
- Discharges (slowly) when the carrier is low
- Time Constant: $f_c = \frac{1}{2\pi RC}\ \mathrm{Hz}$
- Should be small compared to any of our frequencies

Peak Detector Output

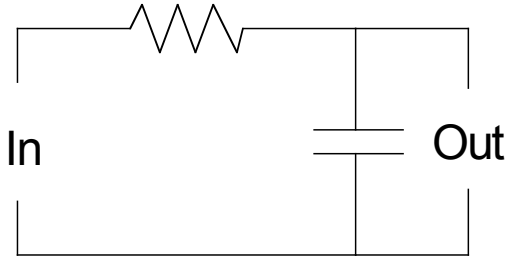
- Circuit:



- Result:



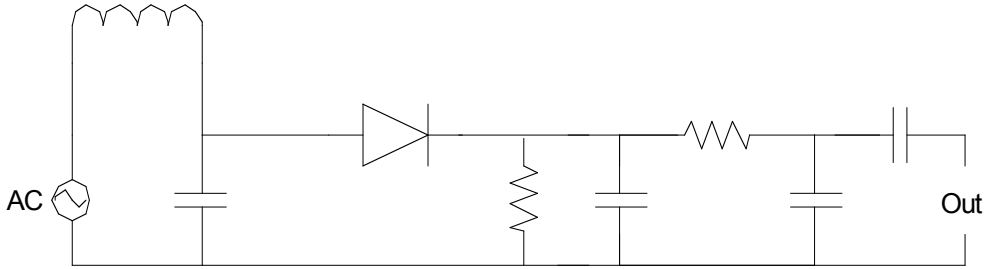
Passive Low-pass Filter



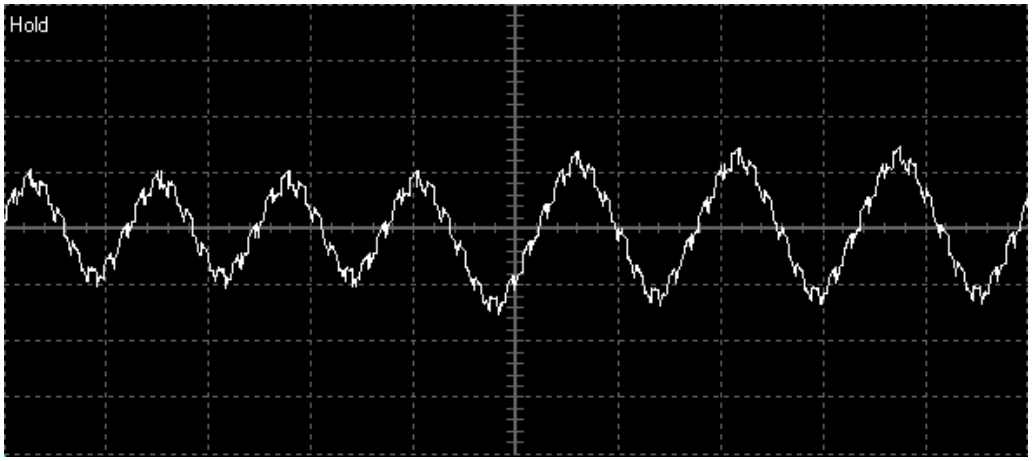
- Again, time constant given by $f_c = \frac{1}{2\pi RC}\ \mathrm{Hz}$
- This is the frequency at which filtering starts
- We want to filter our 125KHz carrier
- We want to leave our 15.625KHz and 12.5KHz data
- Set the time constant below either of them
  - Level of filtering goes up with frequency

Low-pass Output

- Circuit:



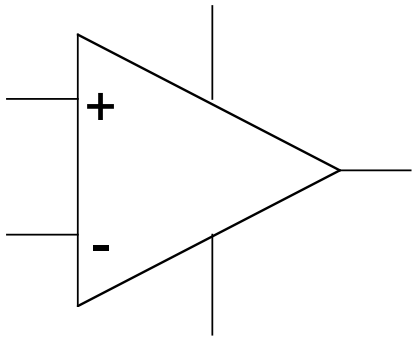
- Result:



- Capacitors are transparent to AC but closed to DC – the final cap removes the (large) DC offset
- Signal is now pretty small – 150mV, no DC

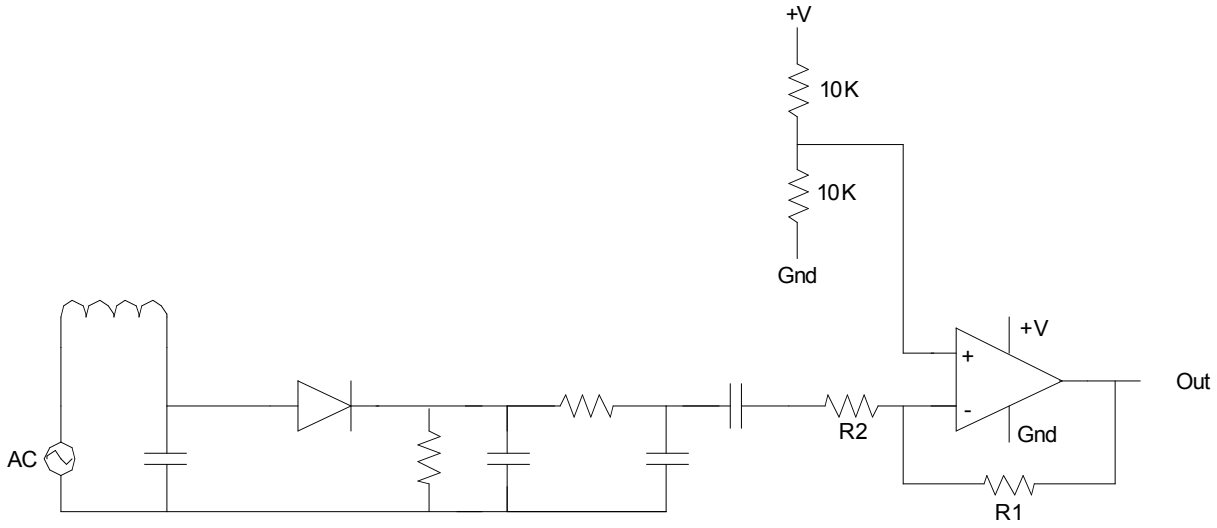
Amplification

- An Operational Amplifier (Op-Amp):



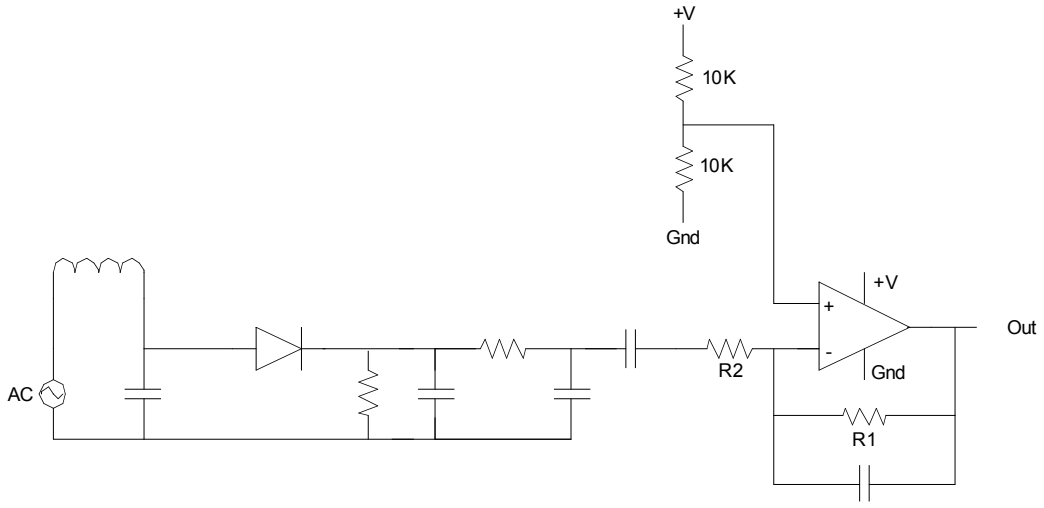
- Subtracts the negative input from the positive input
- Mutiplies the result by infinity
- Produces an output voltage
- They get confusing very quickly
  - Plenty of sample circuits are available

Amplification



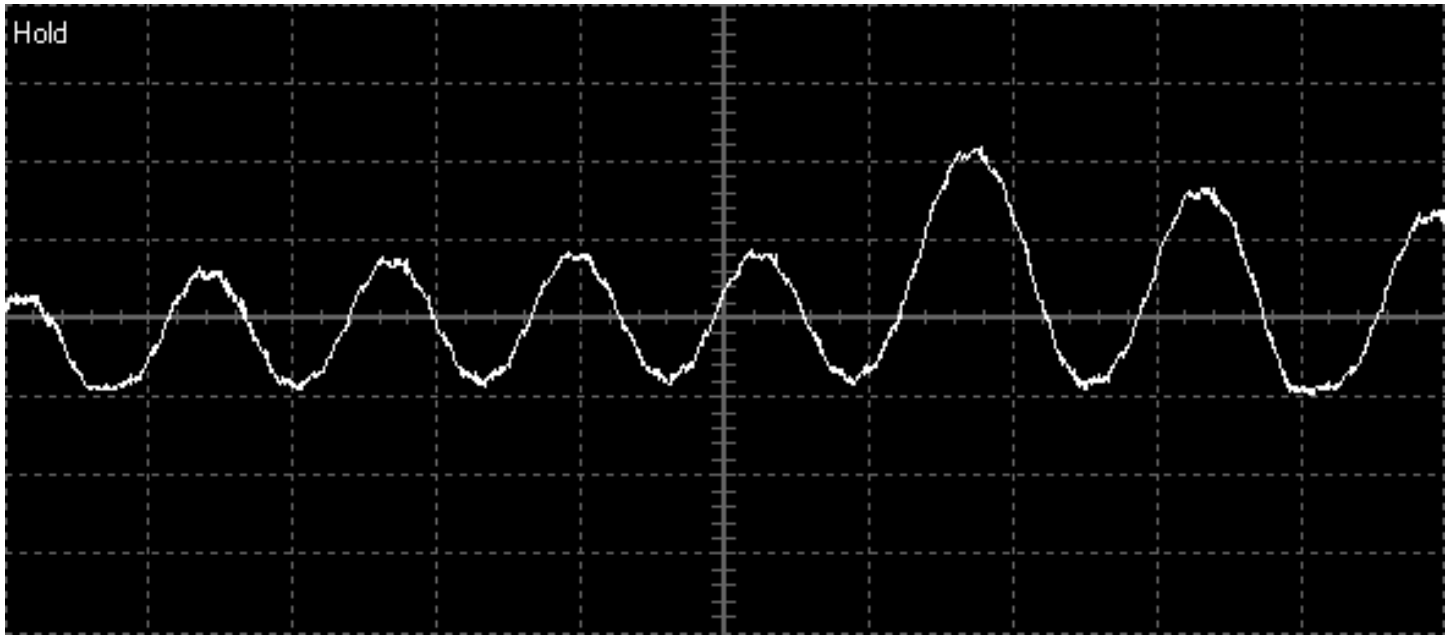
- Output voltage is:
  - Centered on 2.5v
    - Handy since we're running off 5V rails
  - Considerable larger
    - Gain is roughly R2/R1
- R1 = 1M, R2 = 22K, Gain = 45, Vout = 6.75V

Active filtering



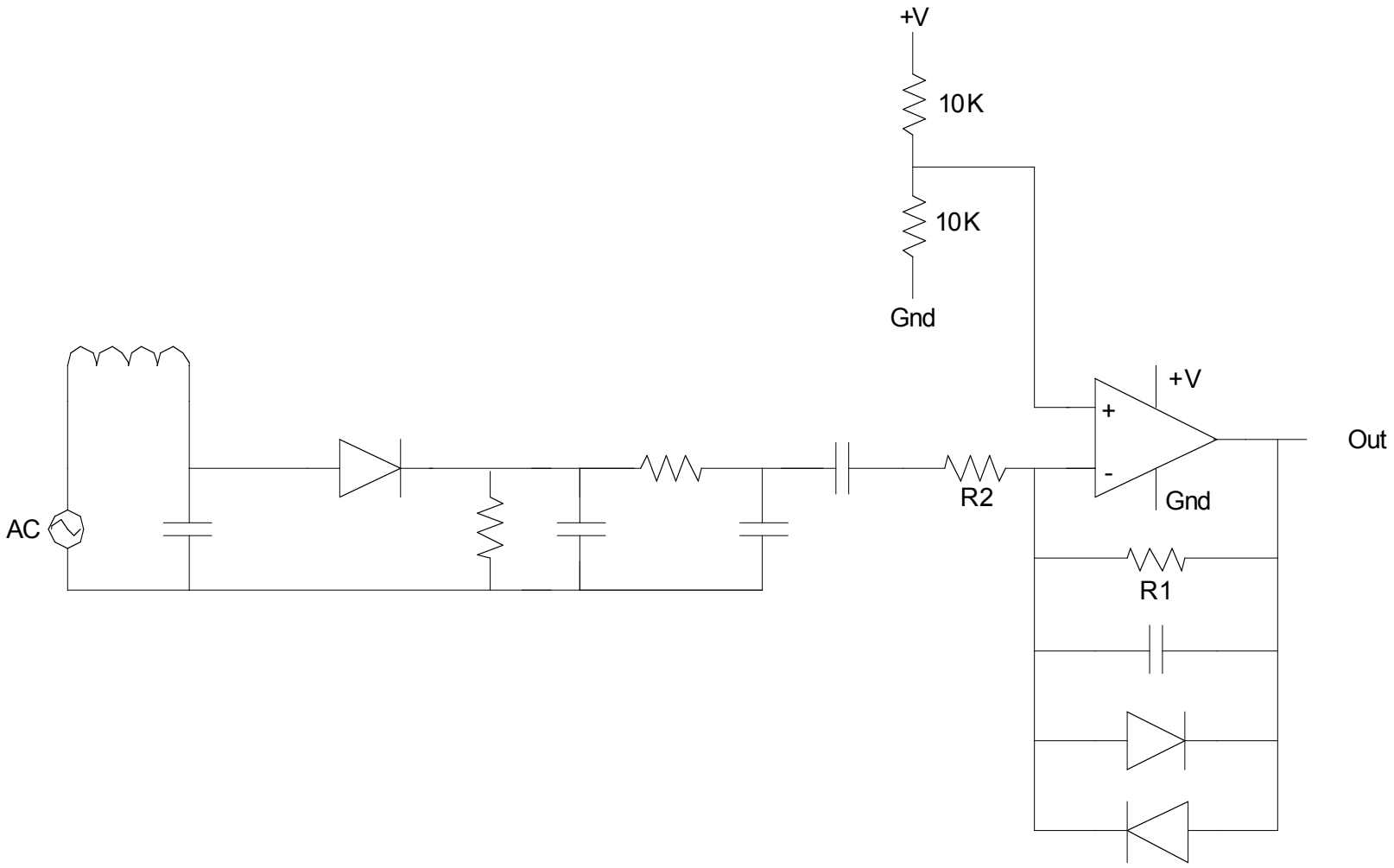
- Capacitor in the feedback loop adds a time constant
  - Frequency dependant == filter
- Same filter characteristics as before
  - Reduces overall gain to stay within supply rails

Active Filter Output



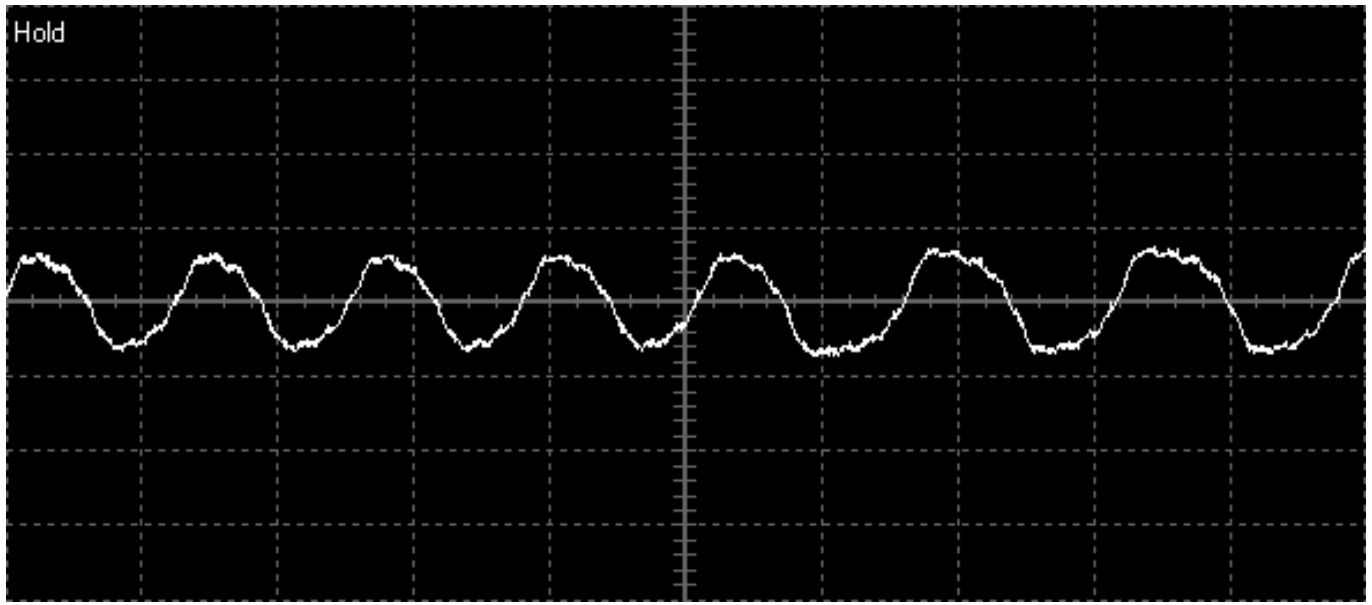
- Carrier is virtually gone
- Some peaks are bigger than others
  - An artifact of the filtering
- Op-amp to the rescue…

Feedback clamping



Feedback clamping

- The diodes restrict how much feedback the circuit can get
  - Gain goes down as voltage goes up
- Result:



We have data!

- We have the demodulated data signal
- We need to decode it into 1's and 0's now
  - Count how long each data wave takes
    - 8 carrier waves, it's a 0
    - 10 carrier waves, it's a 1
- Need some intelligence in the circuit…

Introducing the PIC

- PIC Microcontroller from Microchip
  - Essentially a computer on a chip
- Vast range of different capabilities and sizes
  - Some even include DSP capabilities
- We'll be using the 16F628A
  - Up to 20MHz clock speed
  - 16 I/O pins
  - Two 8-bit and one 16-bit timers
  - PWM output module
  - Two comparators with built-in voltage reference module
  - 2K program memory
  - 224 bytes of SRAM
  - 128 bytes of EEPROM

PIC Programming

- PIC is an unusual environment
  - Very different from a PC
- MPASM language is easy enough
  - Only 35 instructions on the 16F628A
- I/O pins get confusing very quickly
  - Most are multipurpose
  - Download the datasheet, and use it!
- Get used to multitasking
  - NOT multithreading
  - Give the PIC something to do, and carry on execution

Generating a Clock

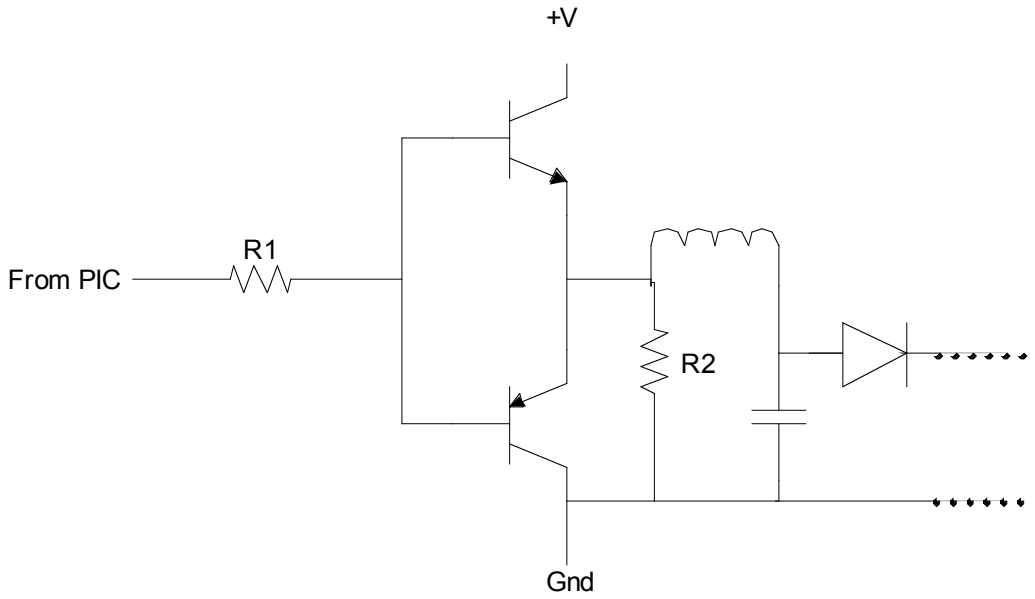
- We need a 125KHz carrier wave
  - We can run the PIC at 20MHz, so should be easy
- PIC takes 4 clock ticks per instruction
  - 5 mips
  - Branches take twice as long
- 40 instructions per carrier cycle
- We can use the PWM module to do the work
  - Why count ticks when the PIC will do it for us?

PWM Output

```
TurnOnPwmPeripheral macro
    banksel PR2
    movlw   39
    movwf   PR2
    banksel CCPR1L
    movlw   20                  ; Pwm duty cycle 50%
    movwf   CCPR1L
    movlw   0x0c                ; Pwm mode, MSBs clear
    movwf   CCP1CON
    bsf     T2CON,      2       ; T2 on
endm
```

- Easy enough – we have our 125KHz carrier now.

Interfacing to the Coil



- Transistors are used as a switch
  - The PIC can’t supply much power
- R2 ensures that some current always flows
  - So that we don’t lose the carrier when faking an ID

Reading the signal (1)

- PIC has two built-in comparators
- Also has a voltage reference
- Set them both up, and feed it the data signal:

```
; Both comparators on, -inputs to RA0/RA1, +inputs to Vreg module
    banksel CMCON
    movlw   0x02
    movwf   CMCON

 ; Set up Vref to 2.5v, the sensed data comes in AC-coupled
 ; about that point.
    banksel VRCON
    movlw   0xAC
    movwf   VRCON
    bcf     TRISA,2
    bsf     TRISA,0
    bsf     TRISA,1
```

Reading the signal (2)

- We're generating the carrier, so we can just count clock ticks
  - Except that we don't have to!
- Use one of the built-in timers to do the counting:

```
; Timer1 enable, internal clock, shut off osc, 1:2 prescale
        movlw   0x11
        movwf   T1CON
```

Reading the signal (3)

- Gluing it all together:

```
clrf            ReceivedBit
ifset           CMCON,DATA_DETECT
goto $-1
ifclear         CMCON,DATA_DETECT
goto $-1
movf            TMR1L,w
clrf            TMR1L
clrf            TMR1H
sublw           0xA8
ifclear         STATUS,C
incf            ReceivedBit,f
return
```

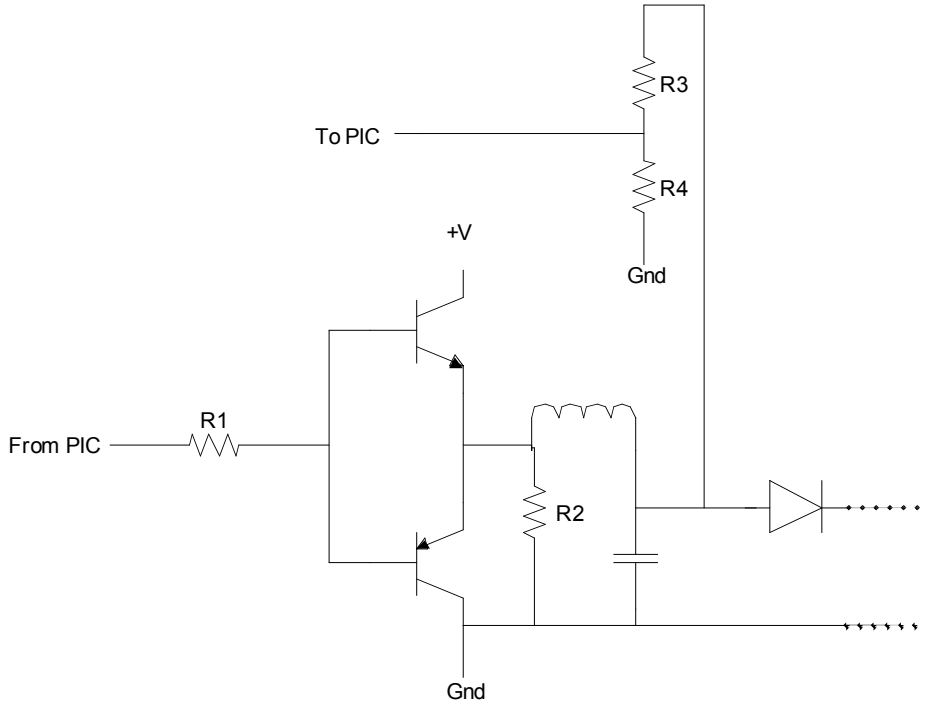
We have the ID!

- We still need some glue code
  - Boring stuff, consult the listing
- At this point, we have the card number in memory
- We could play it back, save it, commit to flash, etc etc.

Playing back the ID

- Easy enough – drive or tri-state the coil pin
  - Drive the pin, enable the transistors
    - Current flows through the coil
  - Tri-state the pin (set it as an input)
    - Transistors are disabled
    - No current through the coil
- Need to listen for the incoming carrier
  - Count cycles, and drive the coil pin in sync

Reading the Carrier



- Incoming carrier could be VERY large
  - 200V?
- Potential divider output voltage:
  - R4 / (R3 + R4)
- Voltage will vary depending on coil efficiency
  - Choose exact values at build time; R3=1M and R4=10K should work

Sending the ID

```
movlw       4
movwf       CycleCount
ifset       TxBit,0
incfCycleCount,f
banksel     TRISB   ;; Open the coil
bcf      TRISB, PORTB_COIL_DRIVER
banksel     CMCON
ifset       CMCON,CARRIER_DETECT
goto $-1
ifclear     CMCON,CARRIER_DETECT
goto $-1
decfsz      CycleCount,f
goto TxNextCycleL
banksel     TRISB  ;; Enable the coil driver
bsf      TRISB, PORTB_COIL_DRIVER
```

Easy as…

- That's really all there is to it!
  - Glue code is long and boring
  - Electronics are the hardest part
- Full schematics are in the paper
- Full code listing is in the paper
- IOActive website has the latest versions
  - Minor errors in the handouts
- Feel free to email me…

Conclusion

- There's a small amount of additional complexity, but not much.
  - A USB oscilloscope is insanely useful
- We don't care about the data format
  - Preamble / postamble are handy
  - Don't bother with checksums
    - Receive the code many times instead
- Clearly, RFID is desperately broken
  - Not all implementations, but most
- Simple electronics is all you need
  - This design is simplified, and a little nasty as a result

References

- Jonathan Westhues' VeriChip cloner
  - http://cq.cx/vchdiy.pl
- PIC16F628A datasheet:
  - http://ww1.microchip.com/downloads/en/DeviceDoc/40044E.pdf
- Microchip reference design for an RFID reader
  - http://pe.ece.olin.edu/projects/proxcard/51115e.pdf
- HID protocol documentation
  - http://pe.ece.olin.edu/projects/proxcard/prox.html

IOActive™

COMPREHENSIVE COMPUTER SECURITY SERVICES

Questions?

chris.paget@ioactive.com