Cricket

Tutorial on using cricket location system

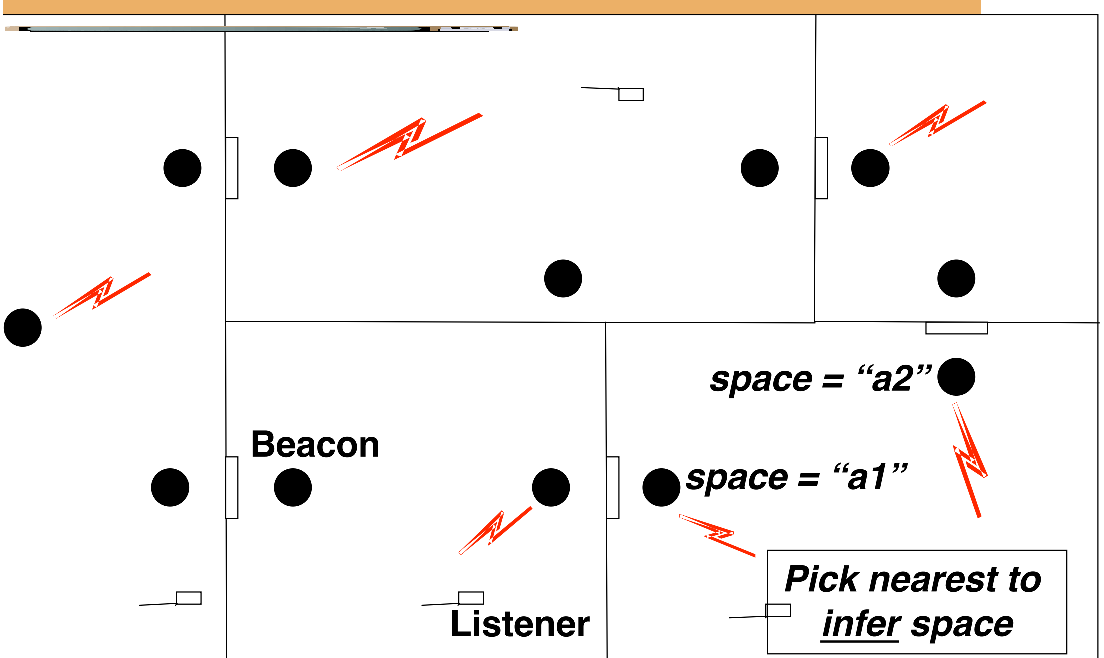
Cricket Goals

- Research prototype
  - build and then evaluate
- Useful mainly indoor environments
  - walls, ceilings not too far
- Recognize spaces, not just physical position
  - good boundary detection is important
    - doors, floors, etc.
- Preserve user's privacy
  - Big-brother can be a bother
    - user has choice to reveal location

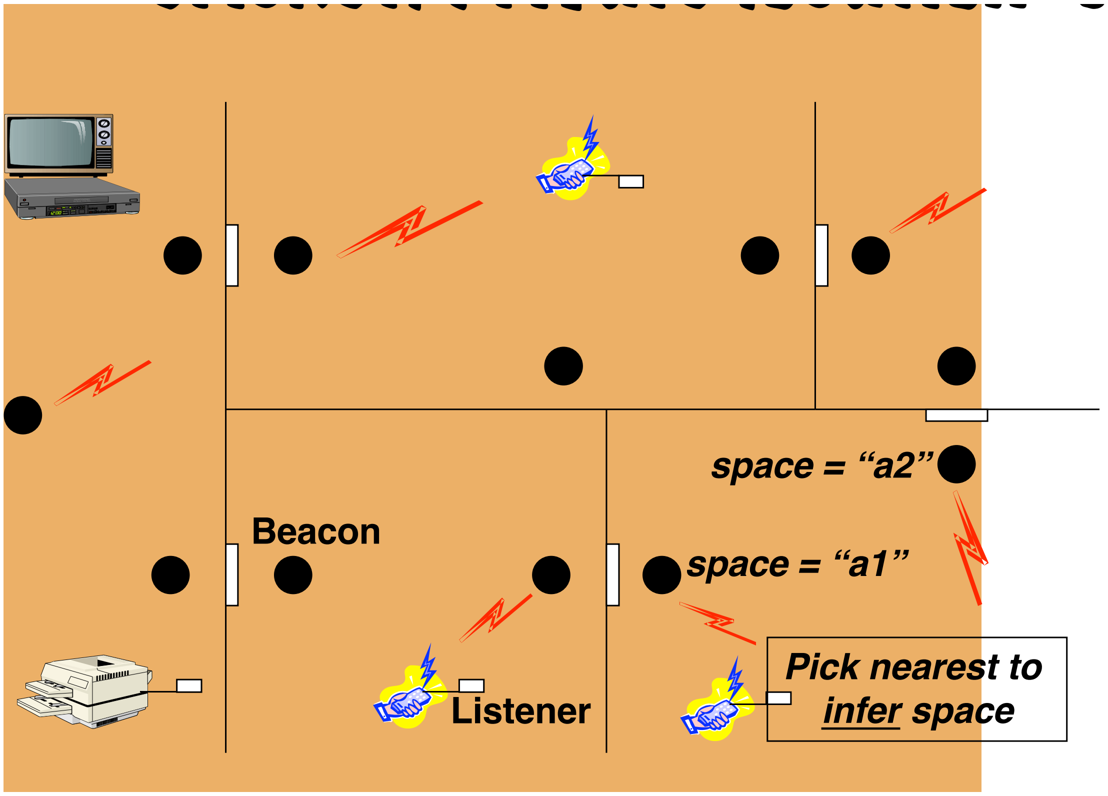
Features

- Distributed architecture
  - No wired infrastructure
  - Easy deployment (no satellites)
  - Low maintenance
- Users are not tracked
  - Listeners are passive
- Integrates with a wide range of resource discovery systems

Cricket: Private location-support



No central beacon control or location database
Passive Listeners + Active Beacons



No central beacon control or location database
Passive Listeners + Active Beacons

Finding the distance

- distance = speed * time
  - want to find the distance
  - we know the speed
- How do we figure out time?
  - Radar: measure round-trip time
    - cannot use it as it violates some goals

Finding the distance

- Synchronized clocks
  - receiver knows exactly when transmitter sent signal
  - how about sending signal first to sync clocks and then 2nd signal?

Finding the distance

- Use two different speed signals
  - both start at same time
  - $d = s_1 * t_1$ $d = s_2 * t_2$
  - We measure delay: $m = t_1 - t_2$

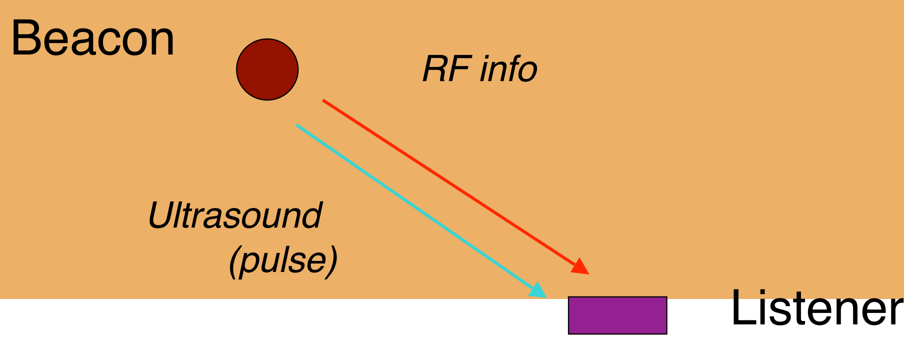
$$t_2 = m * s_1/(s_2 - s_1)$$

$$d = m * s_2 s_1 /(s_2 - s_1)$$

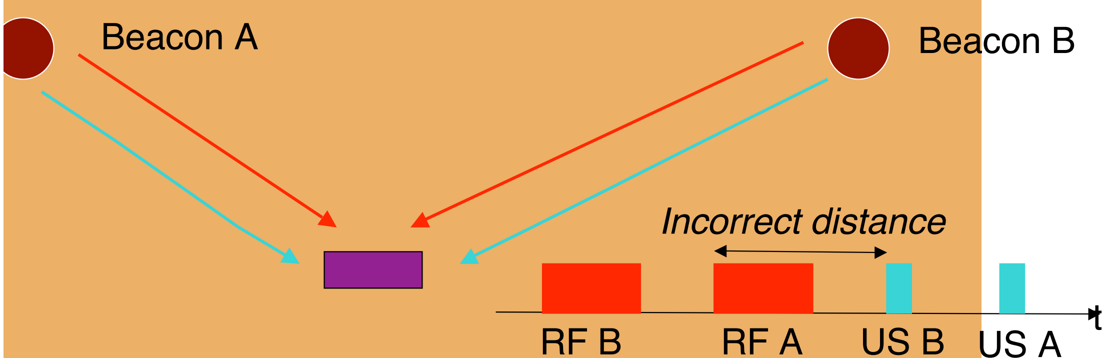
Location Estimation

Distance estimation via coupled RF and ultrasonic signals

- Beacons send information on the RF channel with concurrent ultrasonic pulse



Uncoordinated Beacons

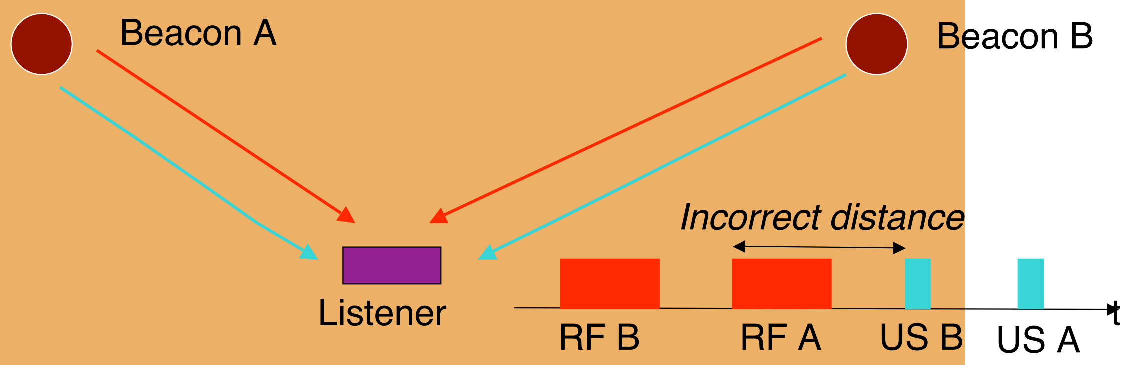


Multiple beacon transmissions are uncoordinated

Different beacon transmissions can interfere

- Causes inaccurate distance measurements at the listener

Multiple Beacons



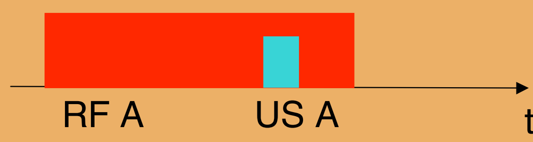
- **Beacon transmissions are uncoordinated**
- **Ultrasonic signals reflect heavily**
- **Ultrasonic signals are pulses (no data)**

These make the correlation problem hard and can lead to incorrect distance estimates

Solution

- Carrier-sense + randomized transmission
  - reduce chance of concurrent beacons
- Bounding stray signal interference
  - envelop all ultrasonic signals with RF
- Listener inference algorithm
  - Processing distance samples to estimate location

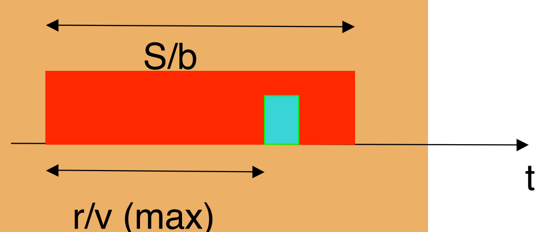
Bounding Stray Signal Interference



- **Engineer RF range to be <u>larger</u> than ultrasonic range**
  – Ensures that if listener can hear ultrasound, corresponding RF will also be heard

Bounding Stray Signal Interference

S = size of space advertisement
b = RF bit rate
r = ultrasound range
v = velocity of ultrasound

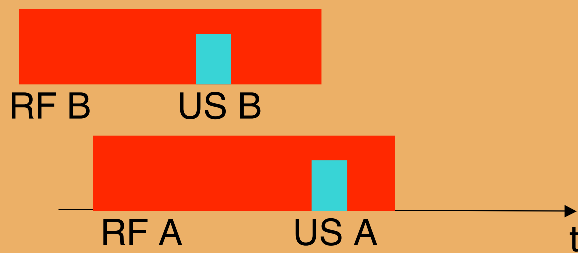


$$\frac{S}{b} > \frac{r}{v}$$

(RF transmission time) (Max. RF-US separation at the listener)

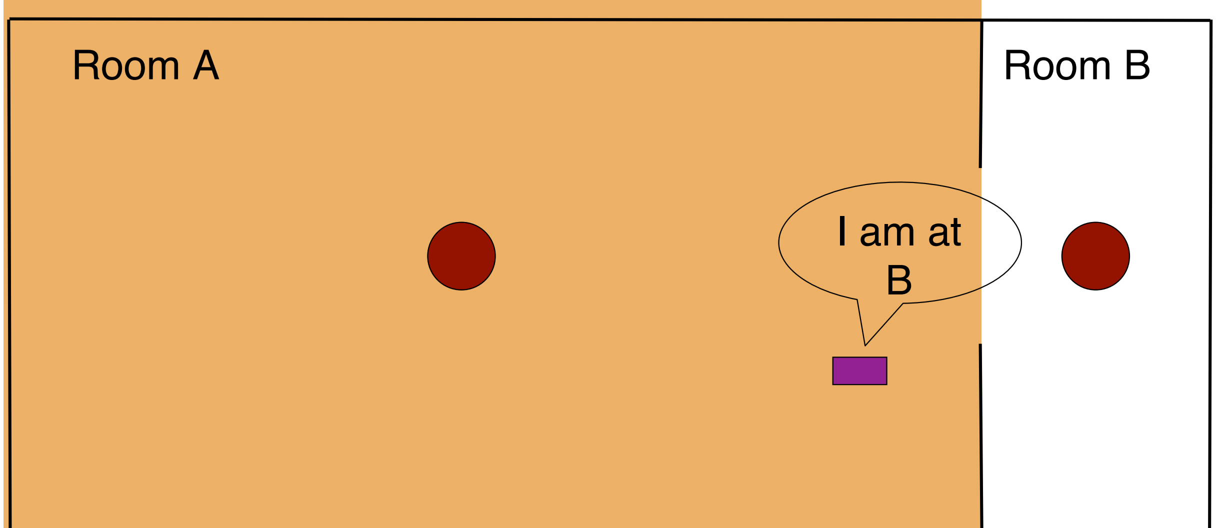
- **No "unaccompanied" ultrasonic signal can be valid!**

Bounding stray signal interferenc

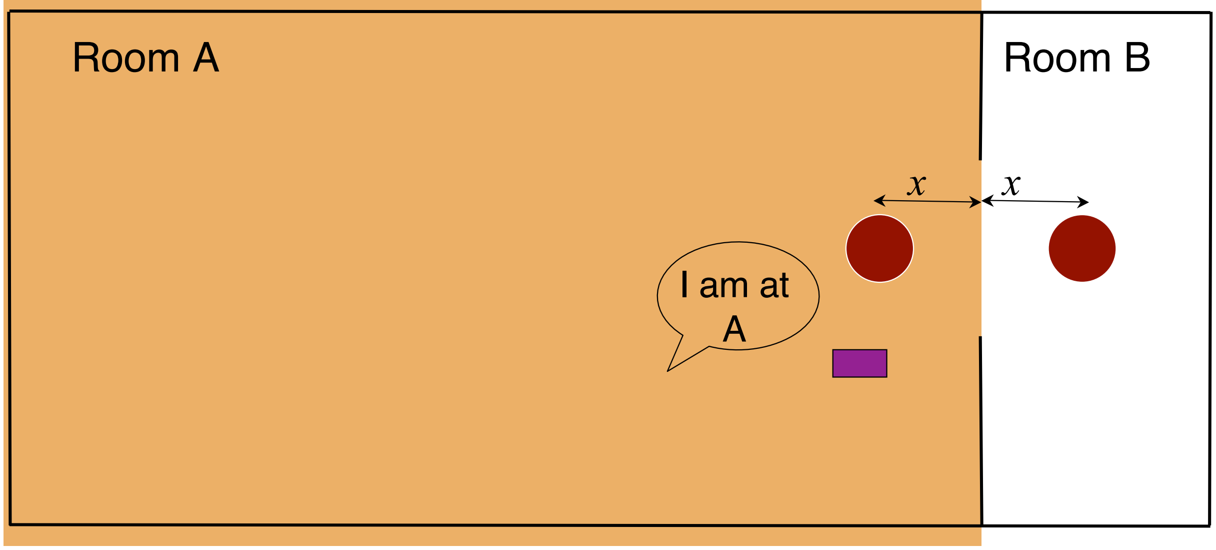


- **Envelop ultrasound by RF**
- **Interfering ultrasound causes RF signals to collide**
- **Listener does a block parity error check**
  - The reading is discarded...

Reflect Correct Space



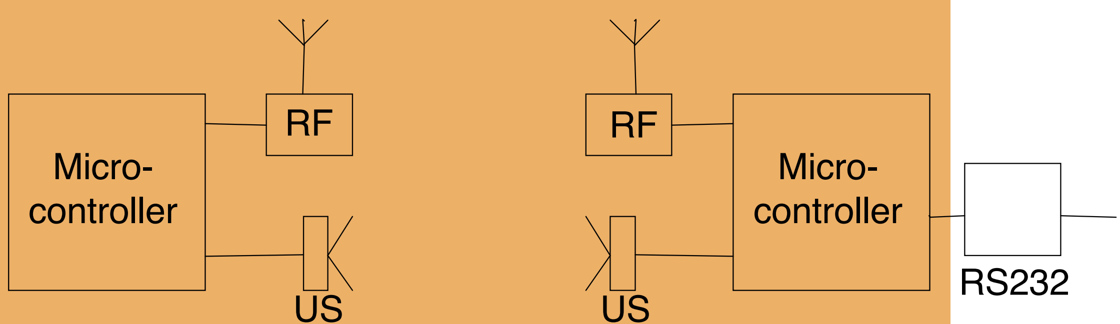
Correct Beacon Placement



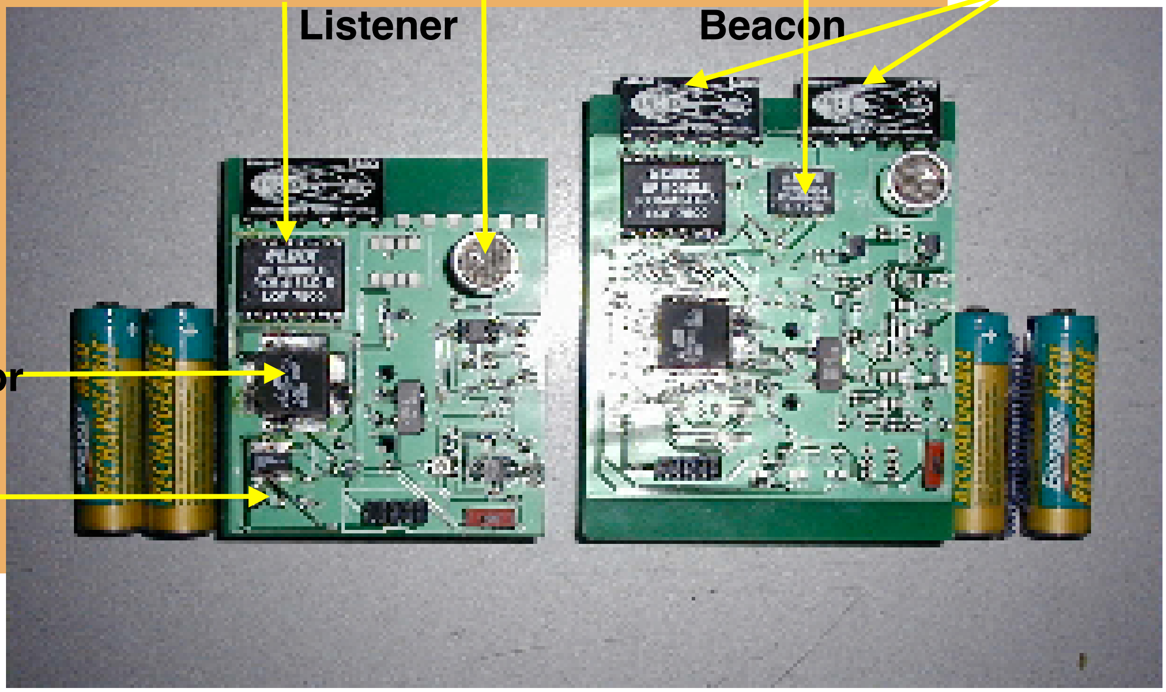
- **Position beacons to detect the boundary**
- **Multiple Beacons per space are possible**

Implementation

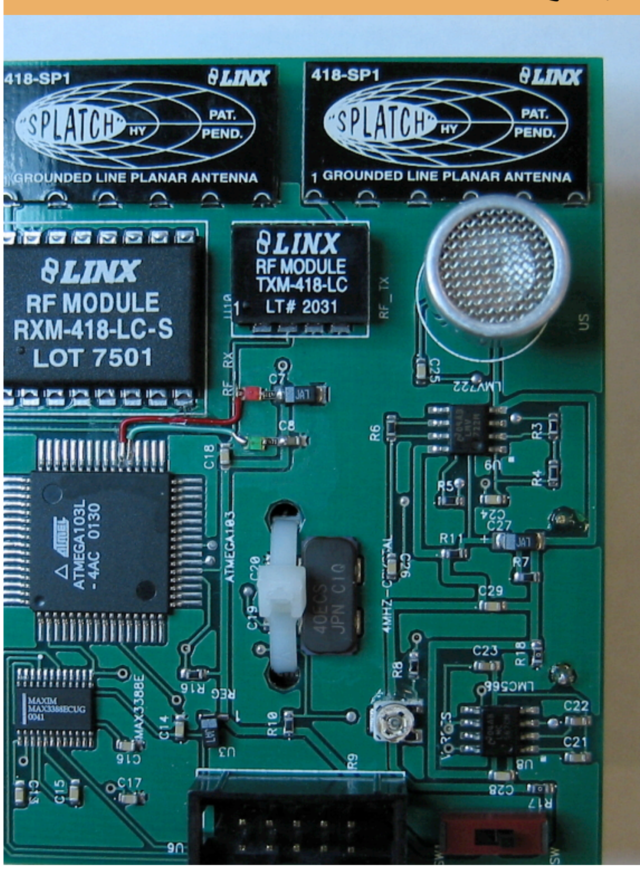
Cricket beacon and listener

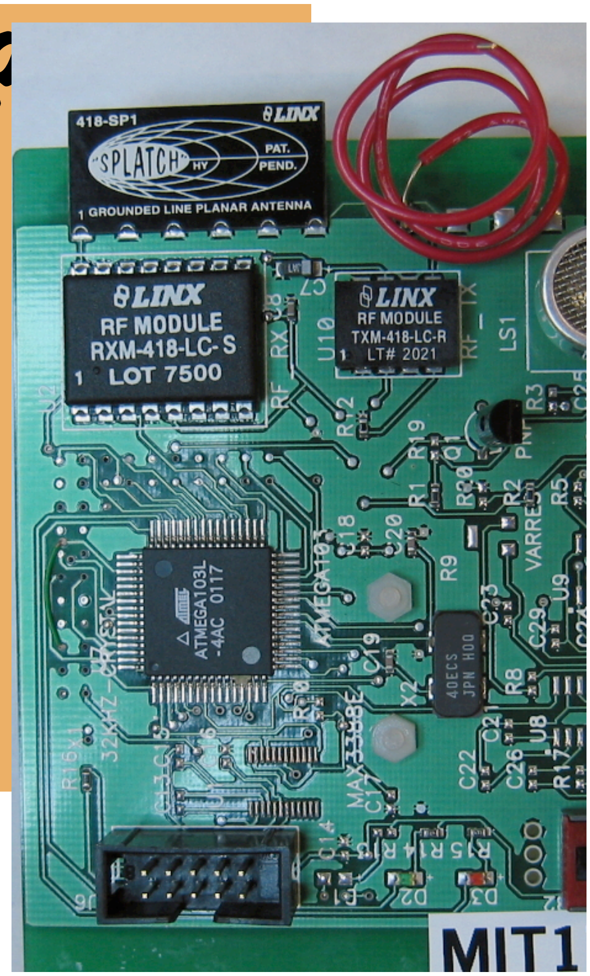


Cricket v1 Prototype

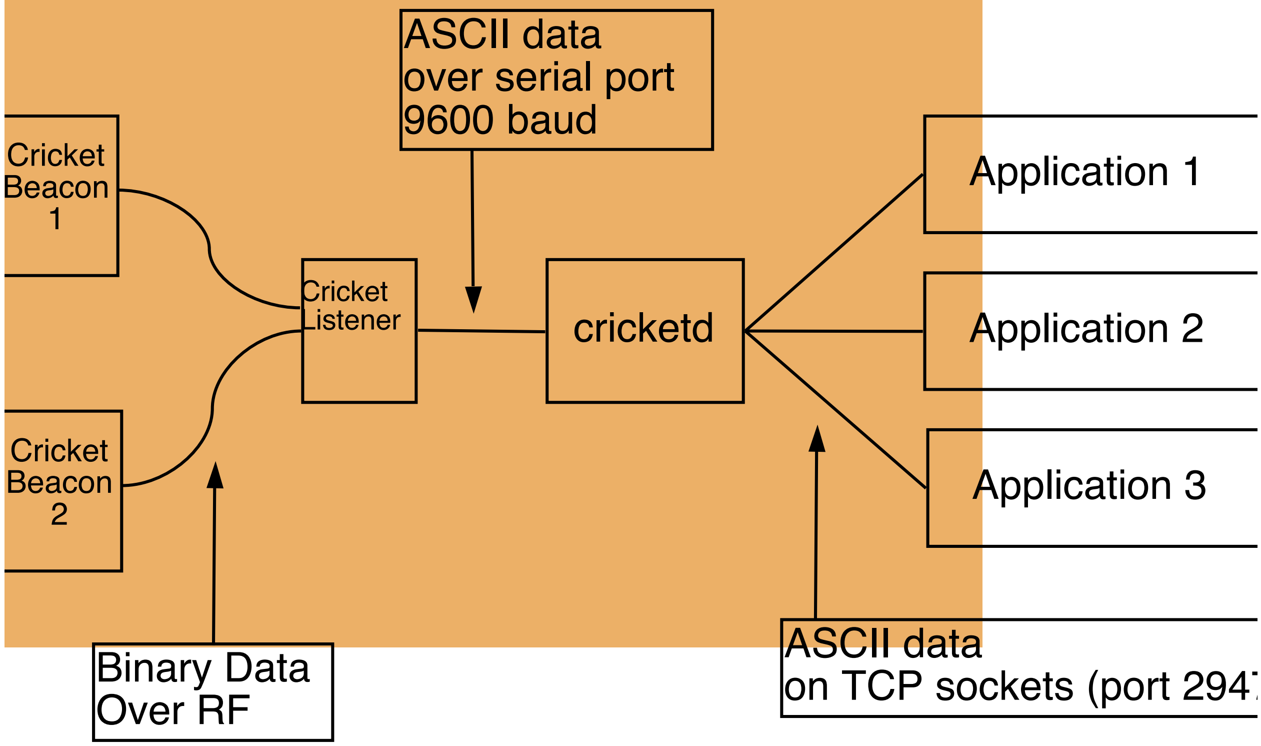


Cricket v1 P





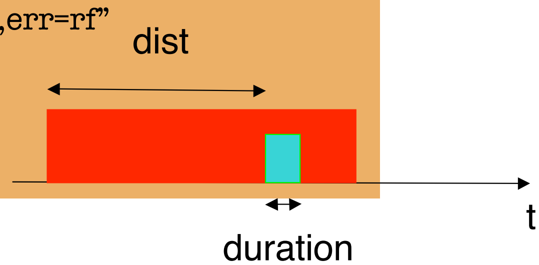
Software Components



Cricket Listener Output

Strings reported from Listeners

- When good RF and good ultrasound pulse heard:
  - "$Cricket2,ver=3.0,space=MIT7,id=20,dist=4F,duration=1A"
- When only good RF heard, no ultrasound heard:
  - "$Cricket2,ver=3.0,space=MIT7,id=20"
- When RF detected, but parity error detected:
  - "$Cricket2,ver=3.0,err=rf"



Speed of Sound

Listener reports distance and duration in 15.625 KH counter cycles (64 microseconds each).

Assume speed of sound is 344.49 m/s then 22.047 mm/cycle

For 343.75 m/s = 22 mm/cycle

Need to subtract 36 units for delay from end of RF to start of US transmission.

Cricket Beacon LEDs

Debug Switch = UP

- Green LED = Transmit
- Red LED = Carried Sensed

Debug Switch = Down

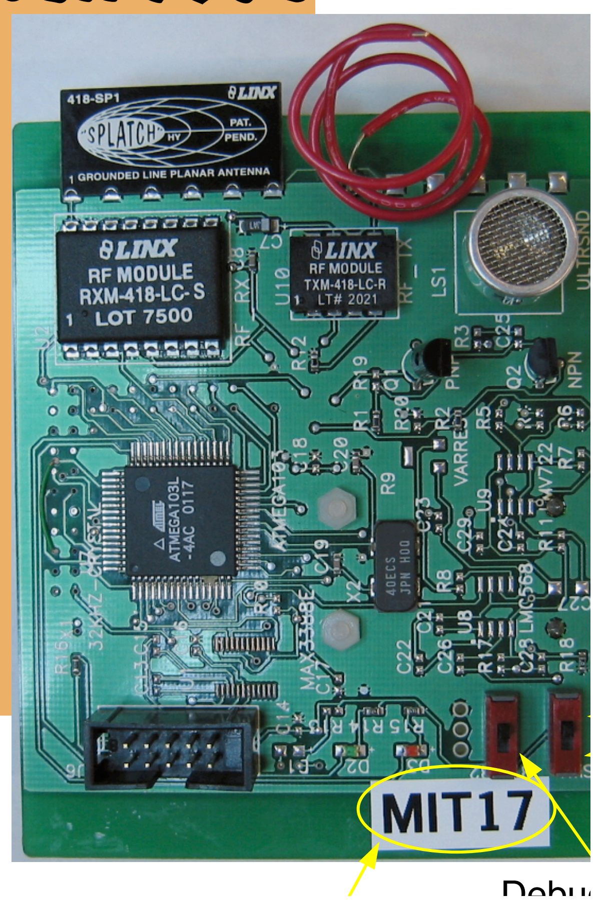
- Green LED = Every 5th transmission

At Startup

- LEDs flash version number
- Red on, Green flash count = Major #
- Green on, Red flash count = Minor #

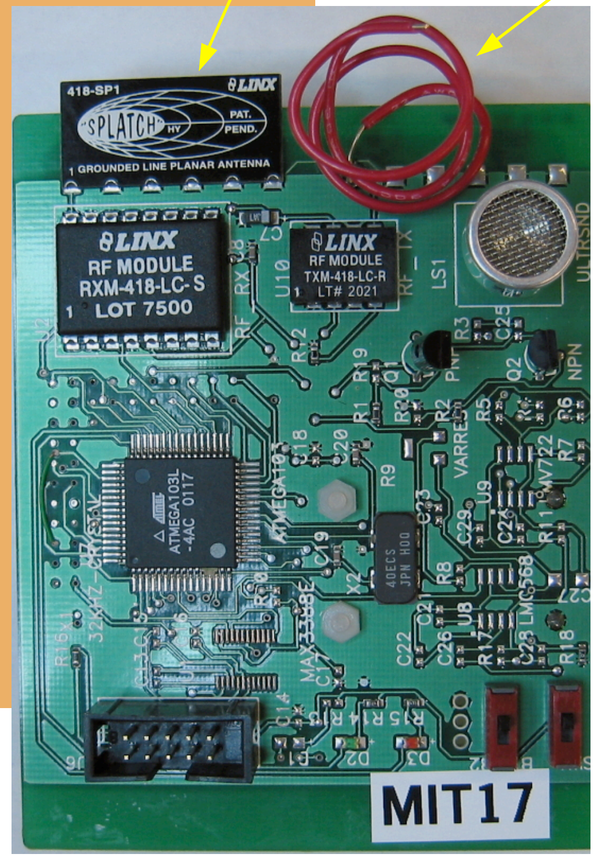
Power Switch

- Up = On



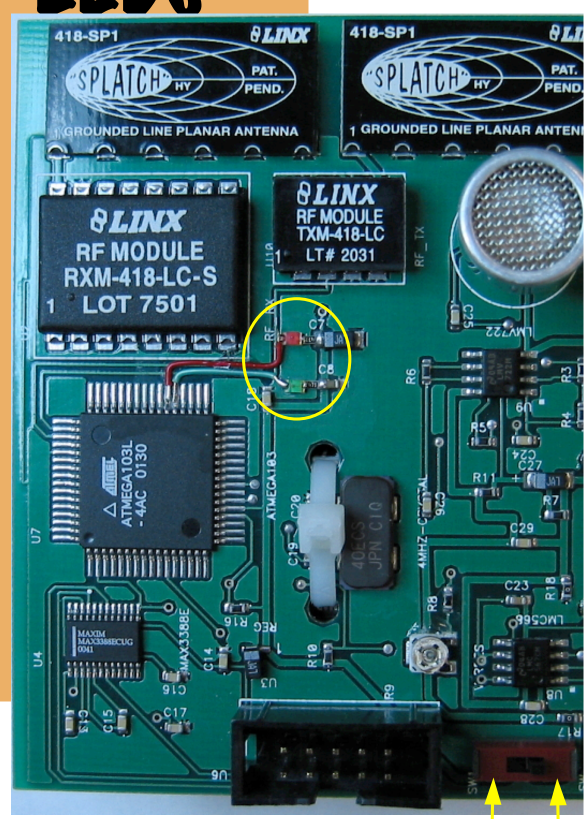
Debu

Cricket Beacon Antennas



- Receive Antenna
  - For sensing transmission of other beacons
- Transmit Antenna
  - Limit transmission distance
  - Should not touch ultrasound
  - Should not cover receive antenna

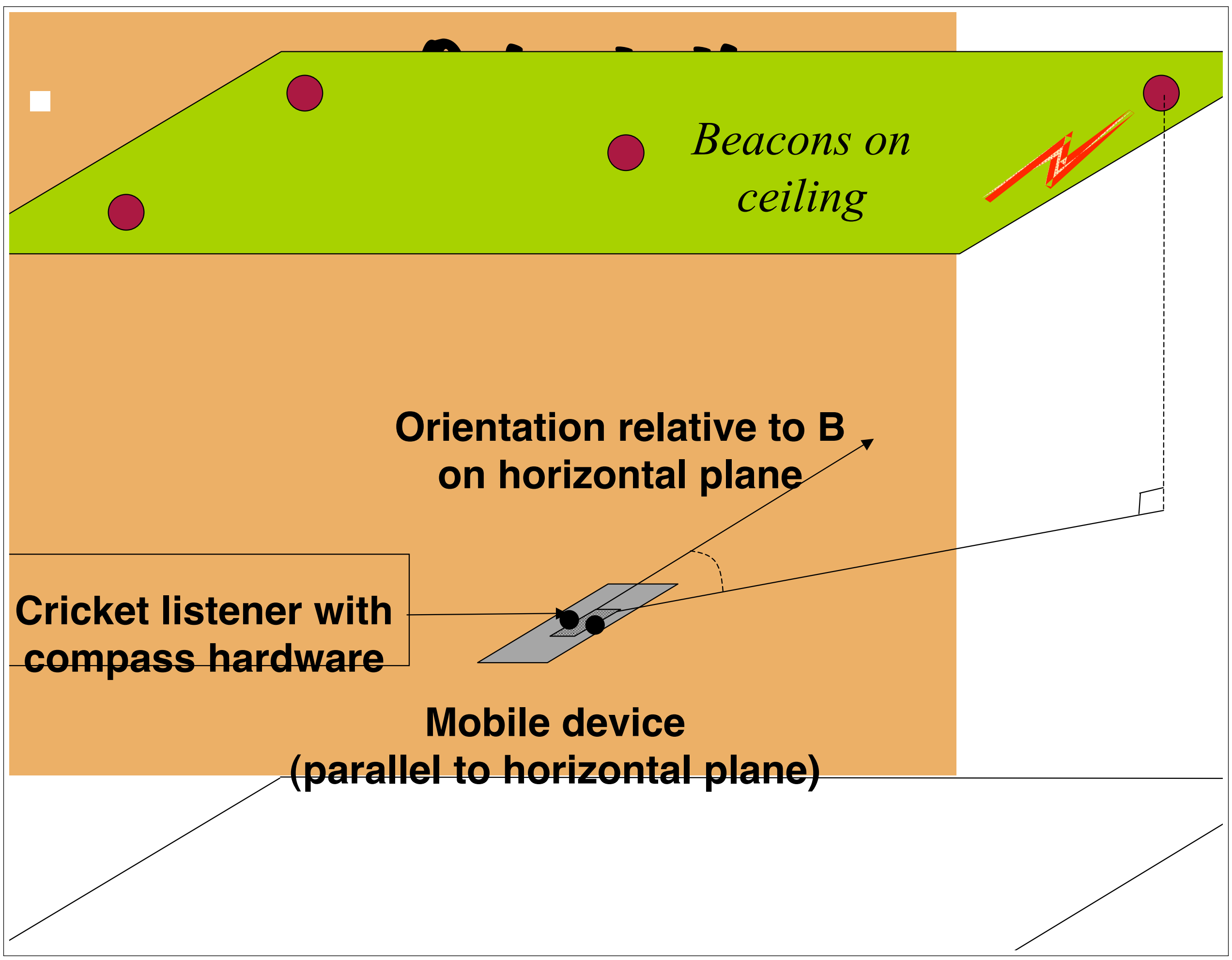
Listener LEDs



- Green Flash
  - Received valid RF and ultrasound
- Red Flash Once
  - Received Radio, but not ultrasound
- Red+Green Flash
  - RF Error (e.g., parity error)
- Red and Green always on
  - Listener not working correctly
- Power On
  - Both LEDs flash together once

cricketd

- Background program (demon) that reads serial port writes data to a socket
- Command line arguments (defaults work correctly on ipaq)
  - -T k Version 3 Listeners (with LEDs) (default)
  - -T c Version 2 Listeners (without LEDs)
  - -S <port> Socket port number (default is 2947)
  - -p <dev> Serial port device name (default "/dev/ttySA0")
  - -s <baud> Baud rate of serial port (default is 9600)
  - -h Help
  - -D <num> Debug level



Hardware Design

- http://nms.lcs.mit.edu/projects/cricket
- http://nms.lcs.mit.edu/cricket/fab
  - User: cricket-fab
  - Password: recipe
- http://nms.lcs.mit.edu/cricket/distrib
  - User: cricket
  - Password: locationrules