Bluetooth

Larry Rudolph
Feb 27, 2007

How to find a partner?

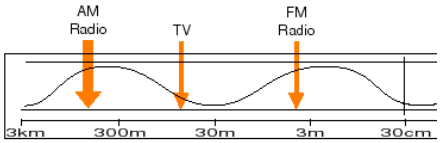
- Why do you want to find a partner?
  - sounds like the Eliza program (theropy)
- Specify needs or services (why should I have to remember some weird name)
  - e.g. "connect me to a nearby color printer"
- Specify location and radius
  - can have nicknames (here or kitchen)

How to find a partner? No, this is not about how to find a boyfriend or girlfriend, but rather how to connect to a particular digital device.

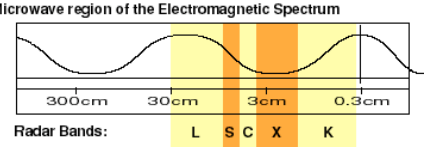
There has been lots of research and many systems built in which one specifies the services required rather than the particular device. Is this a good thing? We generally think so, but it can get complex quickly. Do I want to connect to a laser printer or a bubble jet one? So, maybe "laser" is not a service. Maybe the service is a high-speed printer. Hmmm, high-speed is a relative term, so that is not the right way either. If there are two printers in the room, grandma may say, connect me to the bigger one, or the black one.

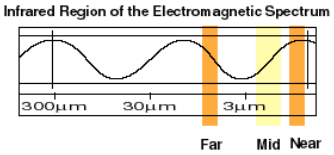
Alternatively, one can specify a location rather than a device.

Wireless Digital Communication



Radio Wave Region of the Electromagnetic Spectrum





- IR (infrared)
  - low cost, low power, directional
  - OK for Remote control -- few bits
  - Failed for communication
    - laptop:printer, laptop:phone

CSAIL

Pervasive Computing MIT 6.883 Spring 2007 Larry Rudolph

There are many ways to communication wirelessly. Infrared (IR) was probably the most pervasive, low cost, low power wireless communication. It was widely deployed, but was mostly a failure for use when sending more than a few bits. Ambient light affected the bit-rate. It just never worked well although it was built into many devices.

Some Standards

- IR (infrared)
  - low cost, low power, directional
  - Short distances (about 1 meter)
  - OK for Remote control -- few bits
  - Failed for communication
    - laptop:printer, laptop:phone
- 802.11 and HomeRF
  - Higher bandwidth
  - more expensive, more power
  - voice not directly supported

Specify by pointing

Specify by IP or Rondezvous

Infrared is a nice technology because it is directional. One can specify the device they want by pointing the remote at the device. Specify by pointing is a big win.

Apple has developed Rondezvous which finds devices on the same wireless sub-net. This is great except that 802.11 takes low power.

In the middle is Bluetooth

- "Bluetooth wireless technology is an open specification for a low-cost, low-power, short-range radio technology for ad-hoc wireless communication of voice and data anywhere in the world"

CSAIL

Not sure to whom to attribute this quote, but there are several interesting aspects to it. One part that we will not address, but is interesting is the open specification aspect. Usually, technologies become open specifications only after several companies have fought it out and there is no clear winner. Or, the company realizes that it has success on its hands and the more products the more they make. Bluetooth was different. It started out as an open standard

History of Bluetooth

- King Harold Blatand, known as Bluetooth, was a Viking and King of Denmark 940-981, who united Denmark and Norway
- 1994 -- Technology was born in an Ericsson study on a wireless technology to link mobile phones and accessories

CSAIL

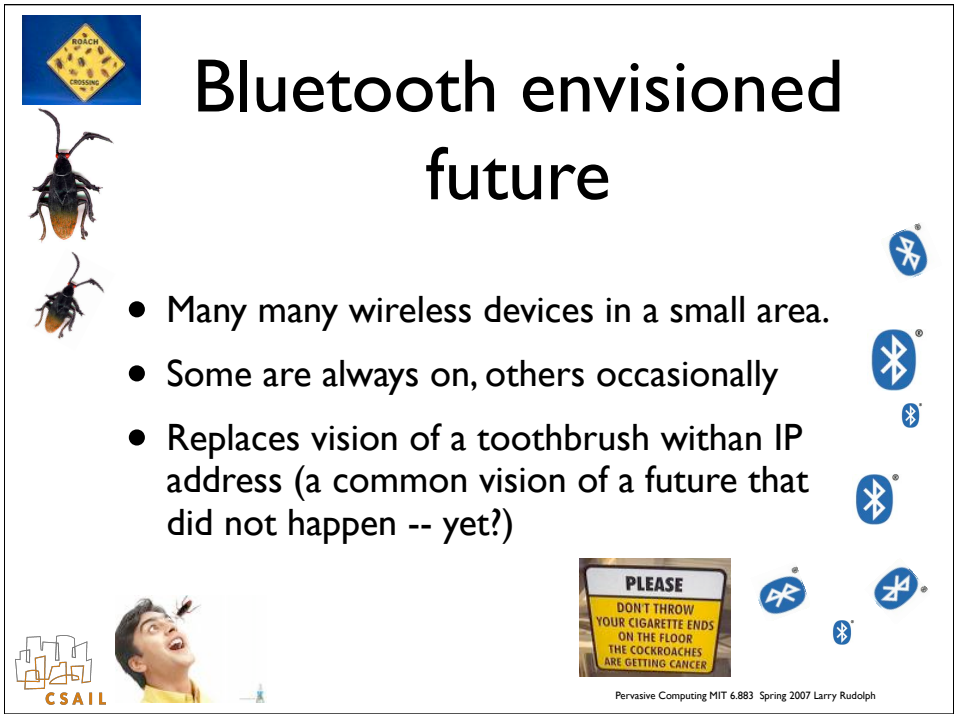
Bluetooth is interesting because it was first develped as an open standard, which is unusual. The typical route is for a company to develop their own, internal standard. Then, if successful, to force others to use it and pay for their use of it. Eventually, after there are several competing standards, the players get together and agree on a single one (or two). Bluetooth was different and it is interesting to see if it was better or worse; more or less expensive.

Bluetooth Vision

- Cable replacement, especially USB
- Local area network
- Automatic connecting of local devices
- Ability to blast advertisements at people who are physically near by.
  - may kill success of bluetooth

The last point (ability to blast advertisements) is especially important as it is something that may cause people to not use bluetooth. Enterprise (companies) like the idea that they can blast ads at people that are physically nearby. It is the electronic equivalent of someone handing out coupons on the street corner. It is even better, since the recipient is someone who buys something above the basic units.

Unfortunately, if one keeps their phone open to spam by stores, the phone is also open to spam by others. The smart thing to do is to keep the phone closed.



The designers of bluetooth envisioned an environment in which everything has a digital interface and communicates with the outside world.

Clearly, lights can be automatically controlled, but do they need an ip address? Do you want someone thousands of miles away to be able to control the lights in your house?

Some details

- Unlicensed ISM band centered at 2.45 GHz
  - 79 channels; every 1 MHz (2.420 to 2.498)
- Mostly for devices within 10 meters
- Expect chips to cost $5
- 2001: first retail products (10 million devices)
- 2003 specification 1.2 (70 million devices)
- Today, mostly in phones & headsets

How many bluetooth chips (or headset devices) are in use today? If you find out, please tell me. The ISM is for: Industrial Scientific Medical band. It is unlicensed in most countries and so anyone can broadcast on it.

The goal is to have a single small inexpensive chip along with a battery (and charger) -- and just plop that into a digital device and presto, you are connected.

Some more details

- Interference comes from
  - Wifi (802.11)
  - Microwave ovens
  - other bluetooth devices
- To minimize packet loss
  - Frequency hopping
  - Adaptive power control
  - Short data packets

A blonde went to the appliance store sale and found a bargain. "I want to buy this TV," she told the salesman.

"Sorry, we don't sell to blondes," he replied.

She hurried home and dyed her hair, then came back and again told the salesman, "I would like to buy this TV."

"Sorry, we don't sell to blondes," he replied.

"Darn, he recognized me," she thought. She went for a complete disguise this time, haircut and perm, new outfit, big sunglasses, then waited a few days before she again approached the salesman.

"I'd like to buy this TV."

"Sorry, we don't sell to blondes," he replied.

Frustrated, she exclaimed "How do you know I'm a blonde?"

"Because that's a microwave," he replied.

CSAIL

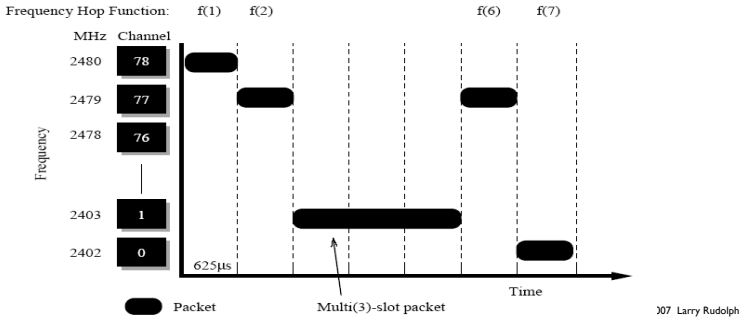
People are afraid to put cell phones near their head because of radiation and so they use a bluetooth headset and stick it in their ear. Not sure why that is better.

There is lots of radio wave interference. Some people envisioned a future in which there may be hundreds of low-power wireless devices in each room of the house. It has not yet come to that and not clear if it will.

Although microwave ovens interfer, I have found that 802.11 interfers with bluetooth and not vica-versa. Bluetooth does not seem to bother 802.11, even though that does not do frequency hopping.

Frequency Hopping

- Big thing about bluetooth is its hopping
  - 1600 hops per second, 625 microsecs / hop
- If there is interference, then just wait
- Makes it hard to eavesdrop (snoop)

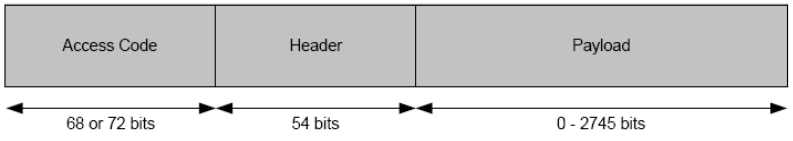


Frequency hopping was invented by the military as a way of avoiding snooping. If one somehow managed to
decode the signal, one would still have to follow along on the right frequencies. My guess is that the broadcaster might also broadcast (garbage) on the other frequencies as well to make it difficult to listen in on all frequencies simultaneously.

In bluetooth, one hops frequencies because some frequencies may have high bit-error-rates.

Power & Packet Size

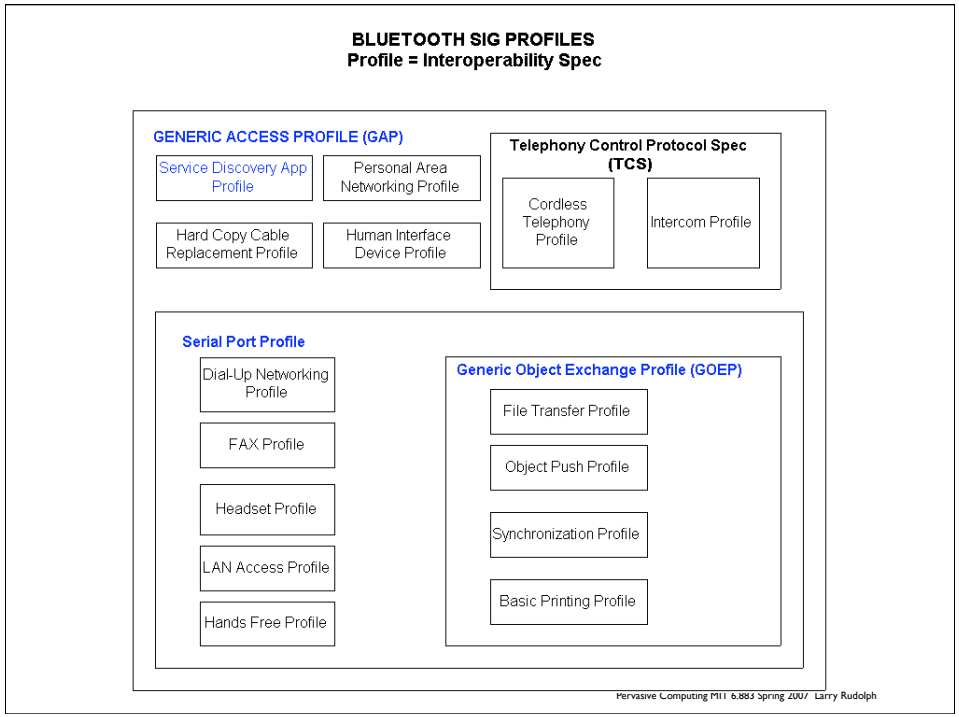
- Three power classes (defined as max power)
  - Classes (1,2,3): 1, 10, 100 mW
- Small packets (compared to ethernet)



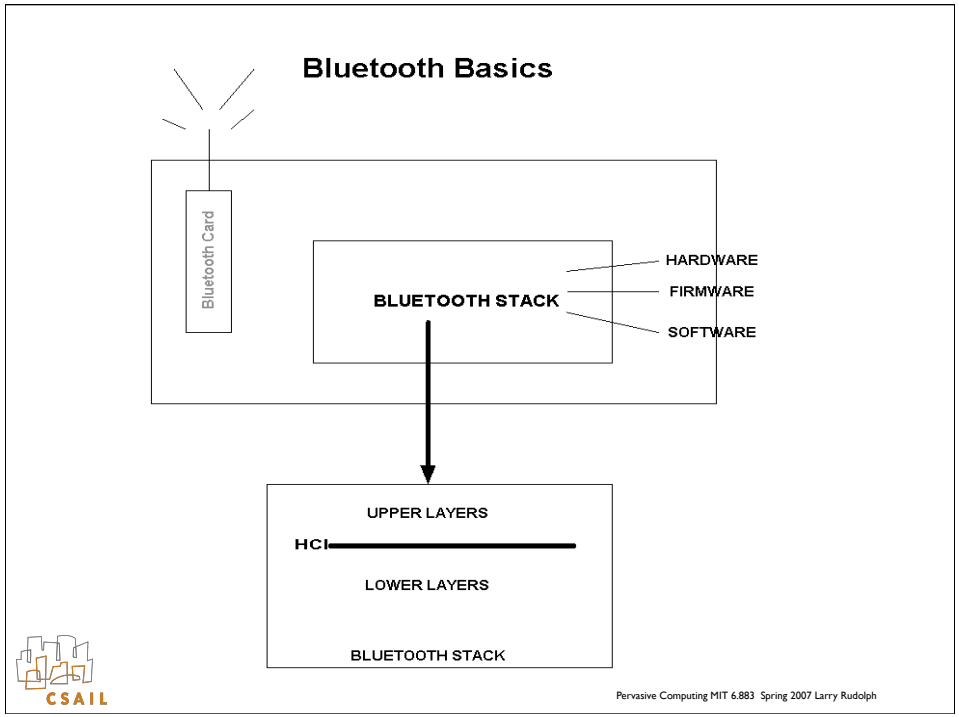
Ethernet has long packets to ensure that if there is a collision, then everyone will hear a collision. If it takes T nano-seconds for a signal to get from one end of ethernet to the other, and if the rate is B bit per nanoseconds, then minimum packet size is B*T bits. For gigabit ethernet, this is large (thousands of bits).

Most BT devices are in the mid-power range. It is via the firmware to vary the power, but that property is usually not publically exported.

Bluetooth is meant to be for short communications and so packet sizes are meant to be short as well.

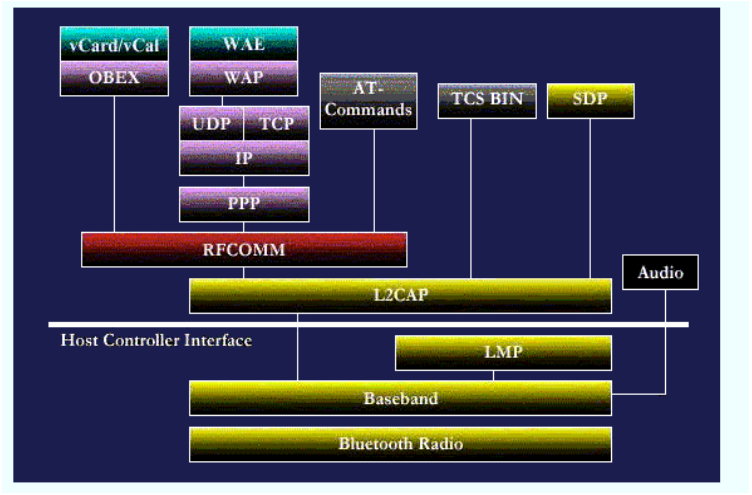


The Bluetooth standard, some 2000 pages long, defines everything from signalling to full protocols.

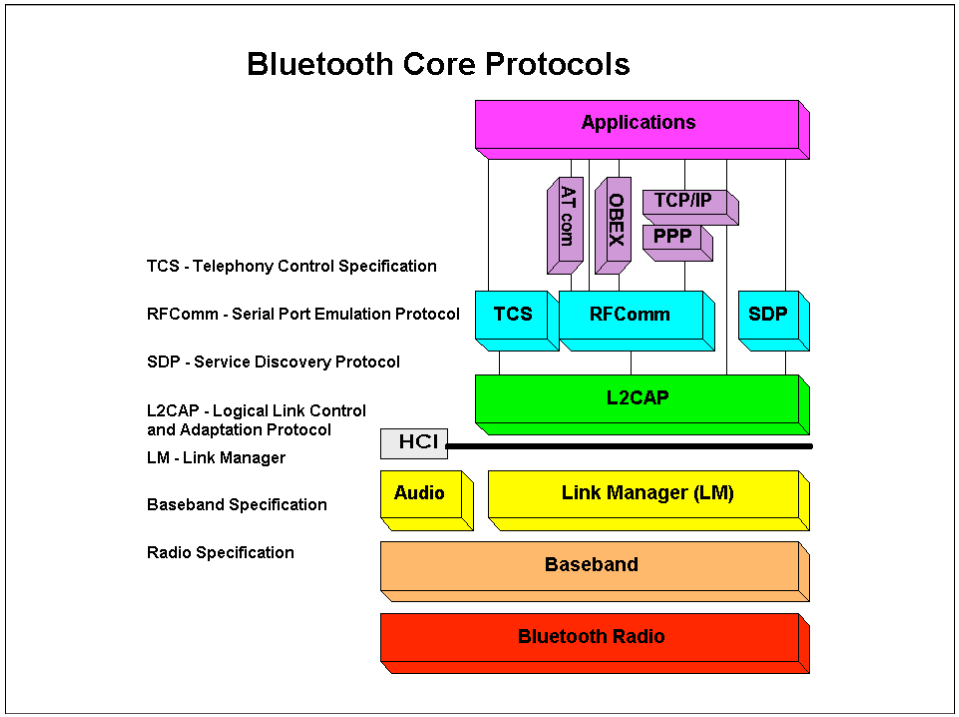


Today, many bluetooth chips also contain embedded microprocessor

Protocol Stack

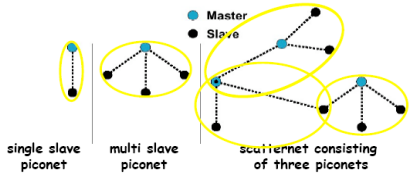


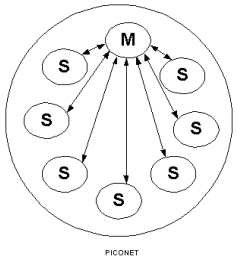
We do not care so much about the stuff under the host controller interface, although that can be
controlled by the HCI commands. The upper level stuff we care less about as well. Audio is completely different, as can be seen from the diagram. Someday soon, video will be there as well, I assume.



Another view of the previous slide, but this shows a nicer layout. But the main point to notice is that Audio is now below the HCI line. In fact the HCI line is not part of the specifications. Each implementation is free to put that line anywhere.

Piconet



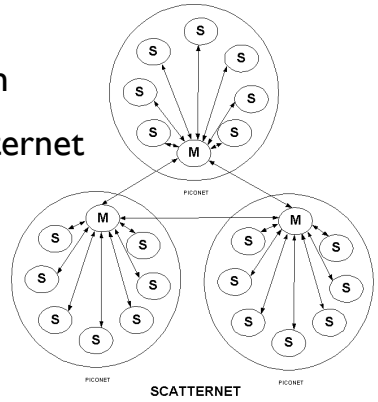


- 8 active devices (master & 7 slaves)
- All hop together
- Master ID and clock determines hop
  - pseudo-random, need seed and start

All Bluetooth docuements make a big deal about Piconets, but it is really not imporant for nearly
anybodies application or usage model.

Scatternet

- inter-piconet communication
- can be 10 piconets in a scatternet
- ad-hoc, P2P network
- (rarely used these days)



Scatternets are even less important.

The Link Manager

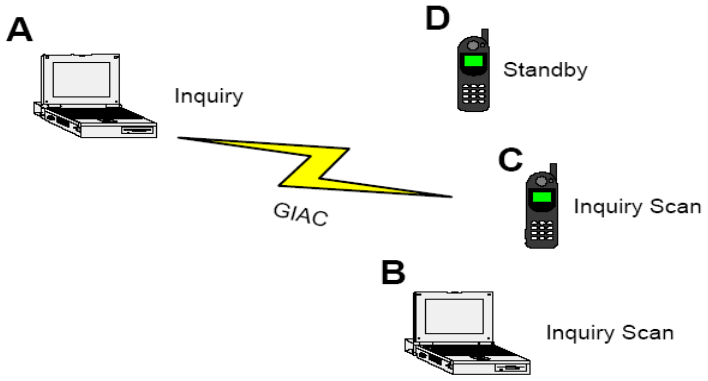
- Responsible for establishing, supervising and tear down connections and logical links.
- Link controller states introduced to carry out these tasks.
- States:
  - Standby
  - Inquiry / Inquiry Scan
  - Page / Page Scan
  - Connection

The link manager, however, is important. The states of the link manager are named in a strange way, so be mindful of it. Note that with TCP/IP one does not really need to know very much about the lower levels of the protocols. Applications using internet access do not know if they are running over ethernet or 802.11. In bluetooth, the lower levels are important.

The Link Manager (cont.)

- Inquiry
  - Used to detect all devcies in an unknown environment.
- Page / Page Scan
  - Describes how connection is established.
  - Have to know the address of the other devices. Is usually achieved through inquiry.
- Connection
  - Master and slaves are synchronized.
  - Connection is established.

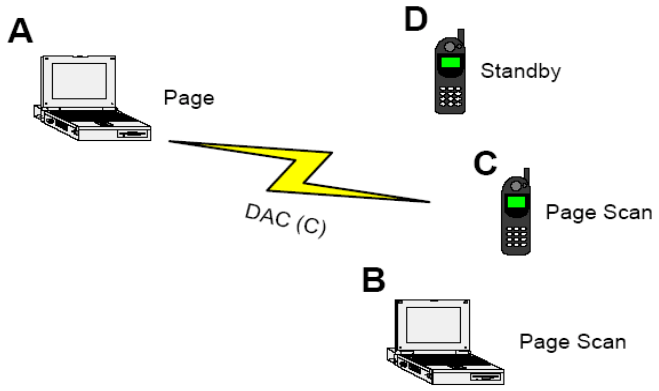
Inquiry / Inquiry Scan



21

A scan

Page / Page Scan



Host Controller Interface (HCI)

- Provided to ease the partition of the Bluetooth Stack across two processors.
- Some systems will implement the baseband and link manager on the Bluetooth device and higher levels on the host processor.
- The HCI is provided as an interface between these parts.

Logical Link Control and Adaption (L2CAP)

- Deals with
  - multiplexing of different services
  - segmentation
  - reassembling of packets
  - Quality of Service

Profiles

- Provide interoperability between devices from different manufacturers for specific services and use cases.
- A profile defines
  - a selection of messages and procedures
  - gives an unambiguous description of communication between two devices.

Bluetooth in ad hoc networks

- Bluetooth network infrastructure is of dynamic ad-hoc type.
- It is constantly changing and depending on the movement of the devices.

Bluetooth in ad hoc networks (cont.)

- Temporary networks.
- Connect ”on-the-fly”.
- Small wireless network called ”personal area network” (PAN).
- Provide voice, data, eliminate cables, bridge networks.
- Supports PDAs, mobile phones, printers, faxes, microphones.

Security

- Bluetooth provides security only over the radio link, from each device to all other devices.
- Three security specifications:
  - Confidentiality
  - Authentication
  - Authorization

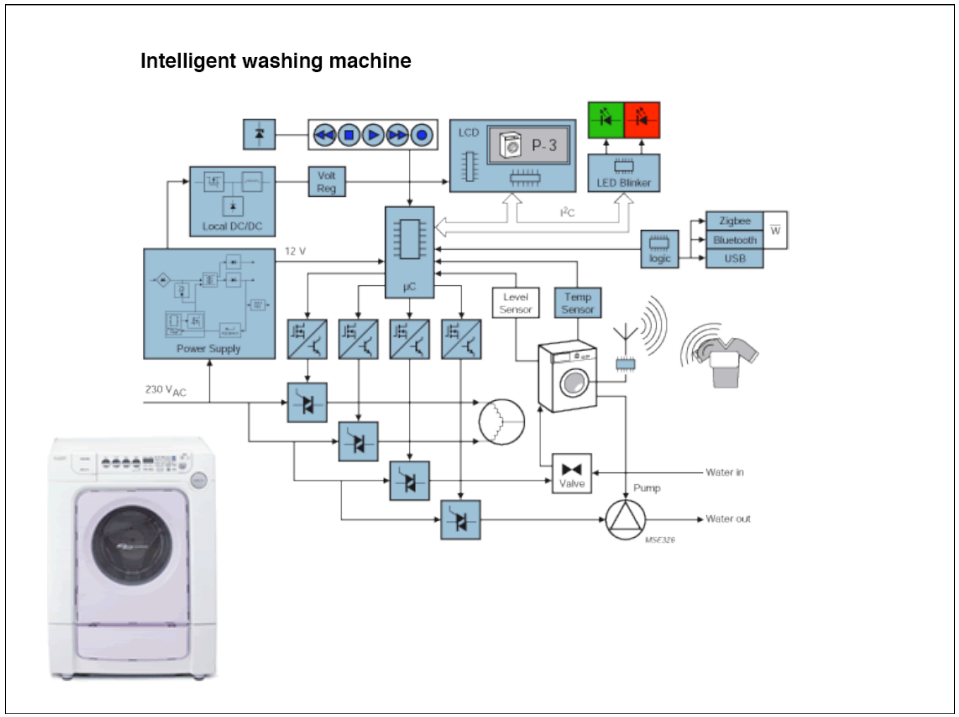
Barbie Bluetooth Headset

Found this on the gizmodo site:

"Some die-hard Trekkie would probably love to get his hands on this. A guy from New Zealand modeled this Barbie doll to a) look like a Star Trek crew member, and b) act as a Bluetooth handsfree headset. Yes, that's right, it's embedded with a Bluetooth headset. According to Ms. Barbie herself:"

You can use me to make and receive calls with Bluetooth 1.1 compatible mobile phones. I have no wires. I work within a 10 meter radius of your phone so you can leave your mobile in your pocket or a bag. You turn me on/off, receive calls, make calls and pair me with other devices by pressing in the small of my back.

CSAIL



30

Why does one want an intelligent washing machine? The washing machine has lots of sensors and knows a lot about the "load". It can "communicate" with the dryer and tell it the state: weight, dryness, capacity, and any special settings. Also, the user need only tell the washing machine information about the contents and not both machines. The machines need not be wired directly.

Many high end machines have particular settings for boys vs girls vs adult clothes. And then there can be additional services added, such as sending an sms message when the wash is finished.

Our focus

- What we don't care about
  - bluetooth headsets, keyboards, ovens
  - how to blast advertisements at users
- What we do care about
  - how to send data between bluetooth devices using python code
- But in this lecture, will give a foundation to let you discover more

Programming Concepts

- Choose a communication partner
- Desired type of communication
- Connection
  - initiate outgoing or accept incoming
- Send & Receive data

Choosing Partner

- Every device has a bluetooth address
  - unlike TCP, same address at all layers
  - 48-bit mac address (unique)
    - could be changed by software
- Devices have bluetooth name
  - User supplied, not unique
  - "My Phone" is a common name

Common protocols

- RFCOMM
  - com port (rs232) replacement, streaming
  - only 30 ports available
  - reliable
- L2CAP
  - connection oriented, customizable reliability
  - reserved ports: 1 -- 4095 (odd numbered)
  - unreserved: 4097 -- 32765 (odd numbered)

Service ID

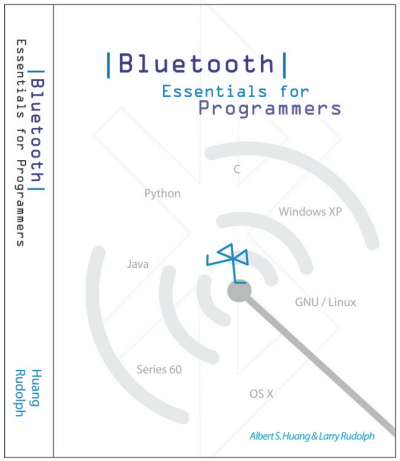
- Every service has a 128 bit (supposedly) unique identifier -- UUID
  - some reserved, developers registered
- Service class ID
- Service Description -- human readable
- Protocol Descriptor -- which prot. used
- Profile Descriptor -- which ones, e.g. mouse

Bluetooth Freq Hopping

- Designed for BT device rich environments
- Lots of radio interference
- Divide spectrum into 72 slices
- Frequency hop between slice
  - pseudo-random hopping
  - hard to track without knowing seed
- Why pairing / discovery takes so long

HCI Tools

- hcitool scan
- hcitool bind
- hcitool connect



Bluetooth Tutorial

Larry Rudolph

CSAIL Massachusetts Institute of Technology

```
from bluetooth import *

target_name = "My Phone"
target_address = None

nearby_devices = discover_devices()

for address in nearby_devices:
        if target_name == lookup_name( address ):
                target_address = address
                break

if target_address is not None:
        print "found target device,address=", target_address
else:
        print "could not find target device nearby"
```

Server (rfcomm/L2CAP)

```
port = 1    #or 0x1001

server_sock=BluetoothSocket( RFCOMM) # or L2CAP
server_sock.bind(  ("" , port)  )
server_sock.listen( 5 )

client_sock, client_info = server_sock.accept()
print "Accepted connection from ", client_info

data = client_sock.recv(1024)
print "received [%s]" % data

client_sock.close()
server_sock.close()
```

Service Discovery

```
port = get_available_port( RFCOMM )

server_sock=BluetoothSocket( RFCOMM )
server_sock.bind(("",port))
server_sock.listen(1)

advertise_service( server_sock, "Bluetooth Serial Port",
                   service_classes = [ SERIAL_PORT_CLASS ],
                   profiles = [ SERIAL_PORT_PROFILE ] )

client_sock, client_info = server_sock.accept()
print "Accepted connection from ", client_info

data = client_sock.recv(1024)
```

```
import sys
from bluetooth import *

service_matches = find_service( name = "Bluetooth Serial
Port",  uuid = SERIAL_PORT_CLASS )

if len(service_matches) == 0:
    print "couldn't find the service!":  sys.exit(0)

first_match = service_matches[0]
port = first_match["port"]
name = first_match["name"]
host = first_match["host"]

print "connecting to ", host

sock=BluetoothSocket( RFCOMM )
sock.connect((host, port))
sock.send("hello!!")
```

Dynamically allocate port

```
from bluetooth import *
socket = BluetoothSocket( RFCOMM )
while True:
    free_port = get_available_port( RFCOMM )
    try:
        socket.bind( ( "", free_port ) )
        break
    except BluetoothError:
        print "couldn't bind to ", free_port

# listen, accept, and the rest of the program...
```

Asynchronous

```
from bluetooth import *
from select import *

class MyDiscoverer(DeviceDiscoverer):
    def pre_inquiry(self):
        self.done = False

    def device_discovered(self, address, device_class, name):
        print "%s - %s" % (address, name)

    def inquiry_complete(self):
        self.done = True

d = MyDiscoverer()
d.find_devices(lookup_names = True)

while True:
    can_read, can_write, has_exc = select( [d], [ ], [ ] )

    if d in can_read:
        d.process_event()

    if d.done: break
```

If confused ...

- Can always go look at source ...
- on my linux machine,
  - /usr/lib/python2.3/site-packages/bluetooth.py
  - look at class DeviceDiscoverer for the skeleton code.