Lecture #25

Bluetooth & CAN

18-348 Embedded System Engineering

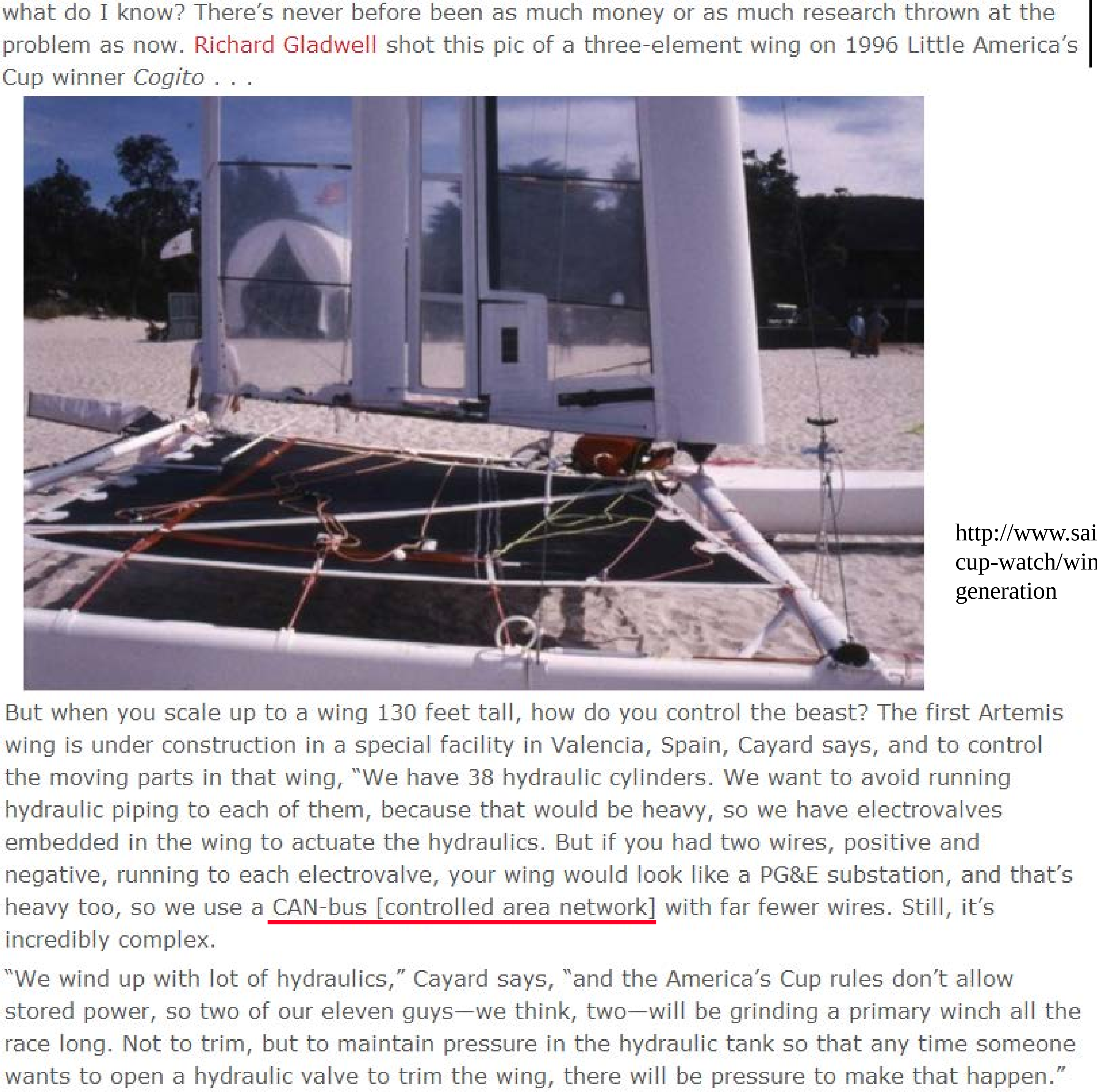
Philip Koopman

Monday, 20-April-2015

(With slides contributed by Chris Szilagyi)

Electrical & Computer ENGINEERING

Carnegie Mellon



what do I know? There's never before been as much money or as much research thrown at the problem as now. Richard Gladwell shot this pic of a three-element wing on 1996 Little America's Cup winner *Cogito* . . .

But when you scale up to a wing 130 feet tall, how do you control the beast? The first Artemis wing is under construction in a special facility in Valencia, Spain, Cayard says, and to control the moving parts in that wing, "We have 38 hydraulic cylinders. We want to avoid running hydraulic piping to each of them, because that would be heavy, so we have electrovalves embedded in the wing to actuate the hydraulics. But if you had two wires, positive and negative, running to each electrovalve, your wing would look like a PG&E substation, and that's heavy too, so we use a CAN-bus [controlled area network] with far fewer wires. Still, it's incredibly complex.

"We wind up with lot of hydraulics," Cayard says, "and the America's Cup rules don't allow stored power, so two of our eleven guys—we think, two—will be grinding a primary winch all the race long. Not to trim, but to maintain pressure in the hydraulic tank so that any time someone wants to open a hydraulic valve to trim the wing, there will be pressure to make that happen."

http://www.sailmagazine.com/cup-watch/wings-next-generation

Where Are We Now?

- **Where we've been:**
  - Control
  - Resets

- **Where we're going today:**
  - Bluetooth
  - CAN

- **Where we're going next:**
  - Test #2 on Wed April 22, 2015
    - Covers all lectures ***after*** Test #1
  - Final projects

Preview

- **Controller Area Network (CAN)**
  - Important automotive network protocol
  - Bit dominance
  - Binary countdown

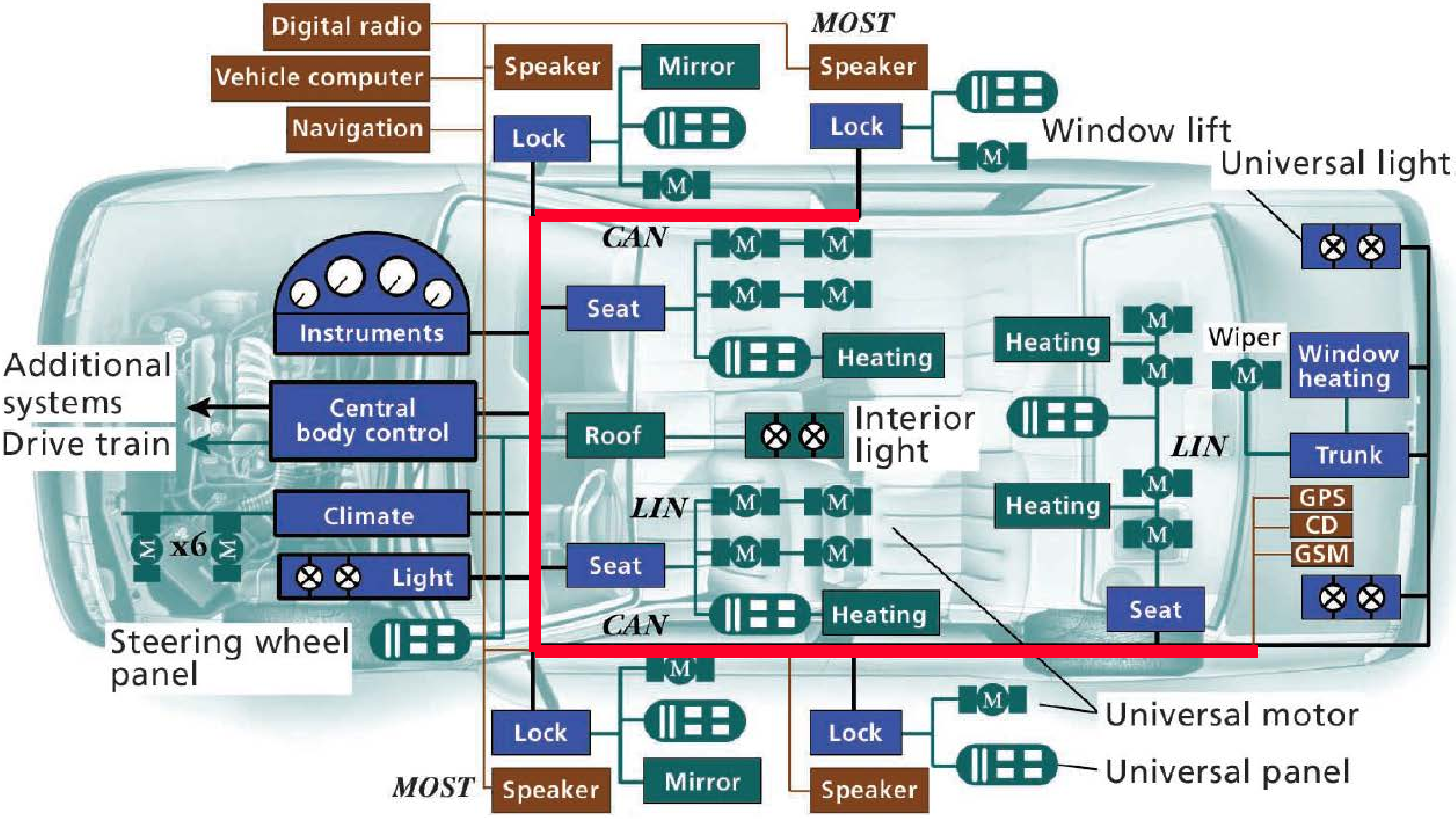
- **Bluetooth**
  - Wireless protocol
  - Gaussian Frequency Shift Keying
  - Frequency Hopping Spread Spectrum

Embedded Communications

- **So far this semester mostly just one microcontroller**
- **What if we want to connect two (or more) computers?**
  - Earlier in the semester you used a serial bus
  - Lab: Communication between module and PC
  - Gets complicated as soon as we connect more than two nodes on a shared wire
    - How do you know which node transmits next?
- **Look at two common embedded network protocols**
  - Wired - Controller Area Network
  - Wireless - Bluetooth

Wired Embedded Network: Controller Area Network (CAN)

CAN Is Central To Automotive Networks



CAN	Controller area network
GPS	Global Positioning System
GSM	Global System for Mobile Communications
LIN	Local interconnect network
MOST	Media-oriented systems transport

[Leen02]

Controller Area Network (CAN) Protocol

- **Originally developed as automotive protocol**
  - Used in essentially all cars made today
  - Real time, high reliability communications among:
    - Controllers
    - Sensors
    - Actuators
    - Human interfaces
  - Up to 1 Mbit/s data rate (for a 40 meter wiring harness)
    - Often run at 250kbps or 500kbps to keep wiring costs down (faster is expensive)

- **The way it is usually used in embedded control networks is:**
  - Each node performs some tasks in its control loop
  - Each node send periodic state data, constantly updating everyone else
    - E.g., “here is the engine speed in RPM” sent at 10 times per second
  - Messages may also indicate infrequent events

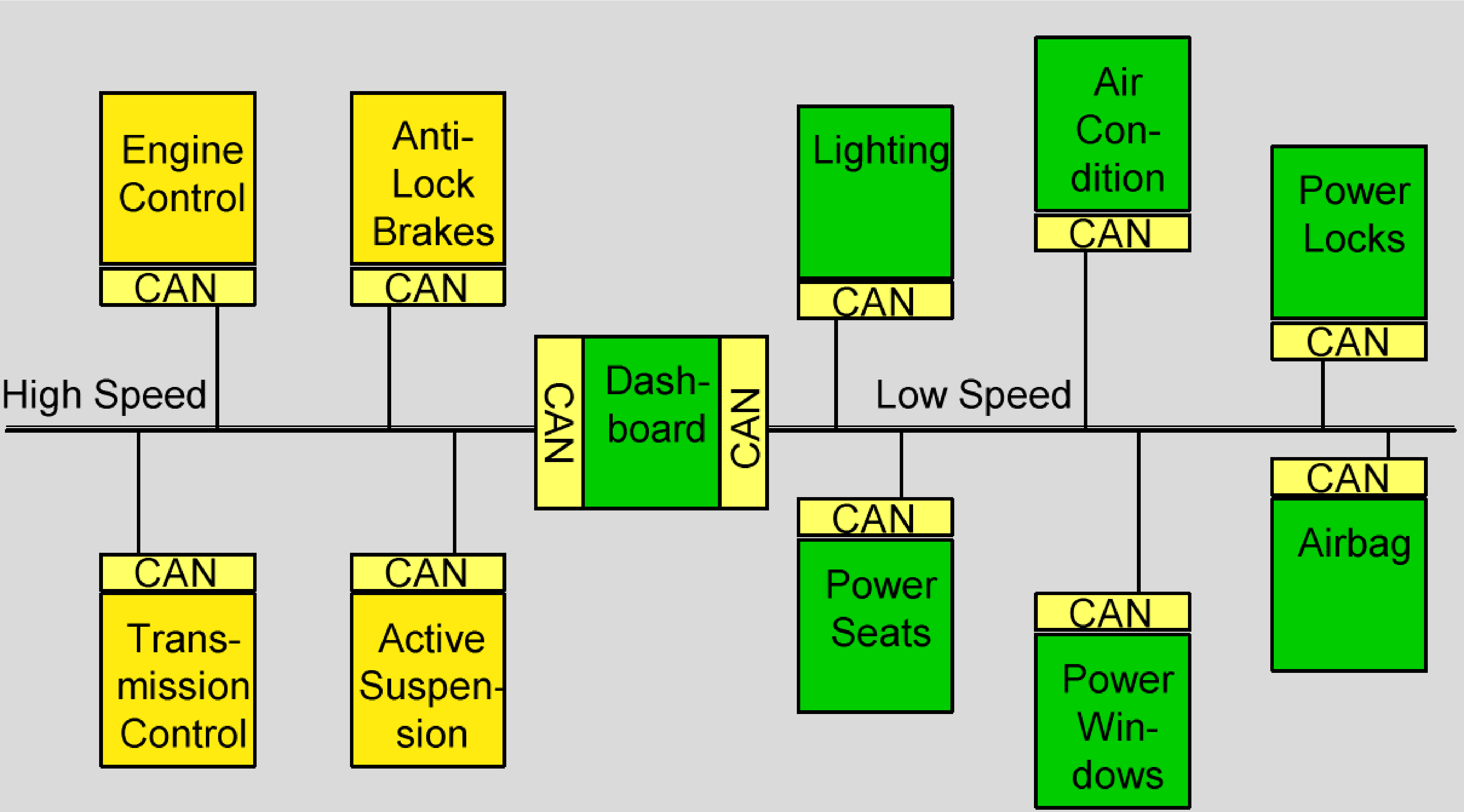
CAN History and Applications

- **Developed and published in 1980s by Robert Bosch GmbH**
  - Originally developed to deal with the complexities of automotive networks
  - Today almost every automobile has at least one CAN network

- **Also used in many other distributed embedded control systems**
  - Trains
  - Ships
  - Industrial automation
  - Robotics
  - Data center power backup systems (coordinates batteries & inverters)

- **One of the most wide-spread embedded network protocols**
  - Anything used in automobiles is forced to be ultra-inexpensive
  - So automotive components are attractive for other applications

CAN – Example Network Topology

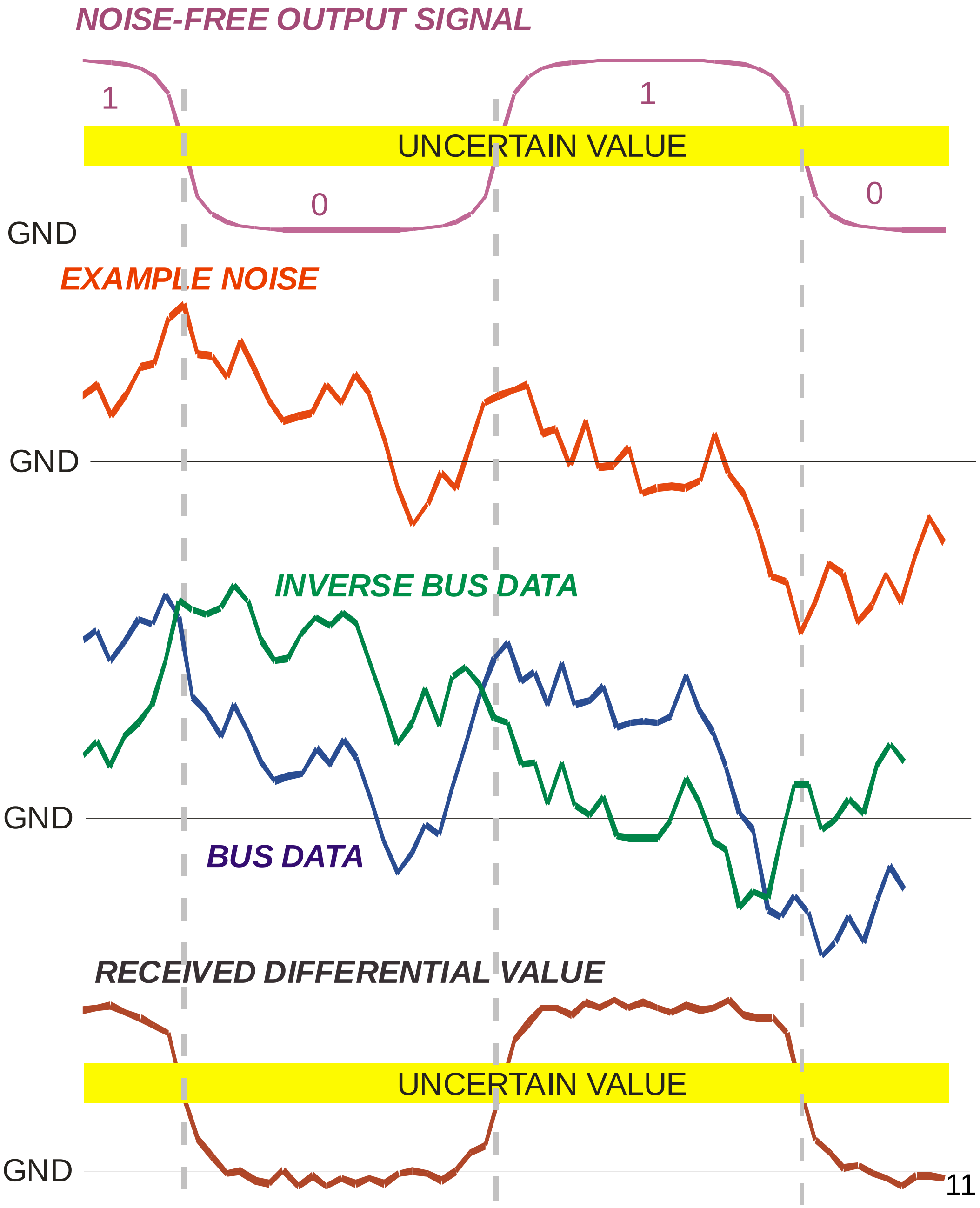


- **Typically set of nodes and topology is fixed at design time**
- **We know exactly what nodes are connected to the network**

[Siemens]

CAN - Physical Layer (Remember this?)

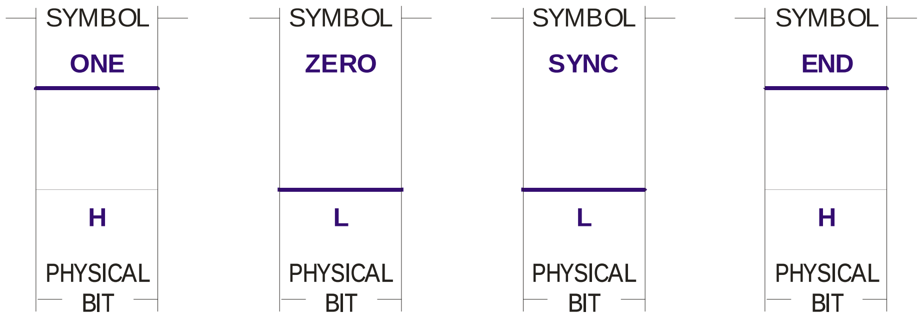
- **Differential Drivers**
- **Send both Data and Inverse Data values on a 2-wire bus**
- **Receiver subtracts two voltages**
  - Eliminates common mode voltage bias
  - Leaves any noise that affects lines differently



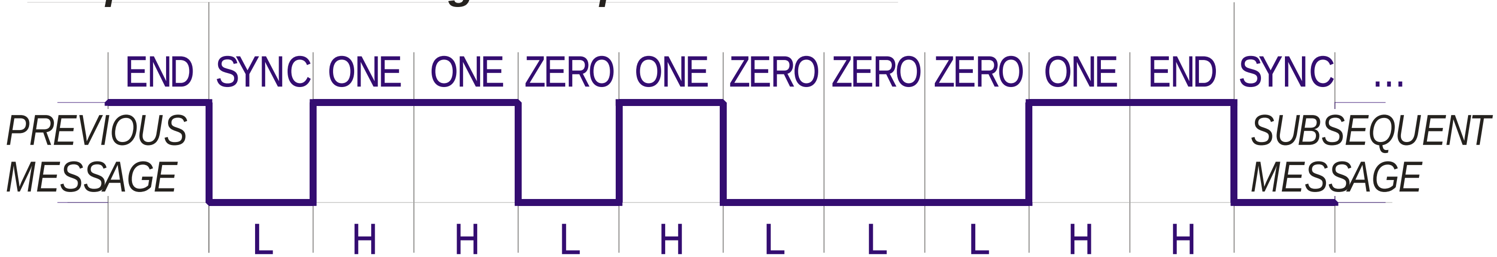
Non-Return to Zero Encoding (Remember this?)

- **Example: Send a Zero as LO; send One as HI**
  - Worst case can have all zero or all one in a message – no edges in data
  - Simplest solution is to limit data length to perhaps 8 bits … BUT
    - CAN uses “bit stuffing” instead of fixed byte size
    - Extra reverse polarity “stuff” bit is inserted after 5 values the same in a row
  - Bandwidth is one edge per bit

Simple NRZ Bit Encoding



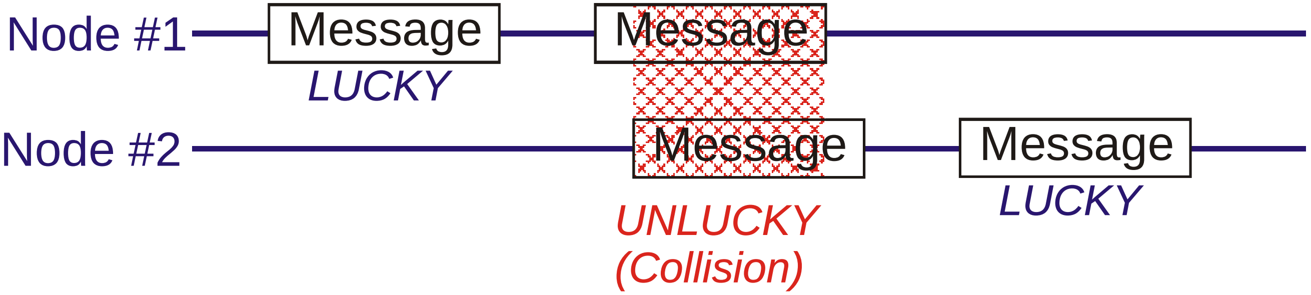
Simple NRZ Encoding Example: 1101 0001



CAN - Media Access Strategy

- **All of these nodes are trying to communicate on the same wire**
  - At some point we're going to get contention on the bus

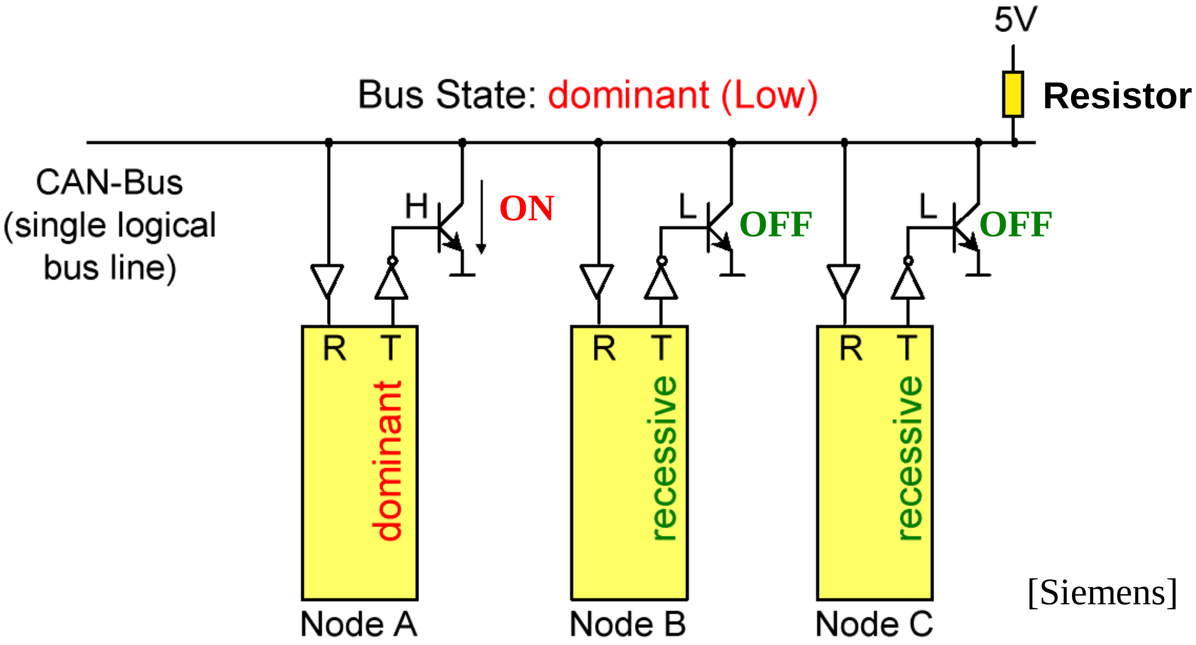
- **What if multiple nodes try to send messages simultaneously?**
  - If two nodes attempt to broadcast at the same time, get a collision



- **Ethernet uses random backoffs then retransmits**
  - But, you can get unlucky and collide repeatedly
  - There are many tricks (see 18-649), but then it is something other than Ethernet
  - Need a ***collision-free*** protocol to make real time deadline guarantees!
- **CAN uses a more elegant solution - Bit dominance**
  - Collision free protocol for real time control – need bound on worst case time

CAN – Bit Dominance

- **CAN uses the idea of recessive and dominant bits**
  - Also called a “Wired OR” design
  - Bus floats high via resistor unless a transmitter pulls it down (down=dominant)
  - (Other bus wire in differential transmission floats low and transmitter pulls up)
- **High is “recessive” value**
  - Sending a “1” can’t override the value seen on the bus
- **Low is “dominant” value**
  - Sending a “0” forces the bus low no matter what another node is sending



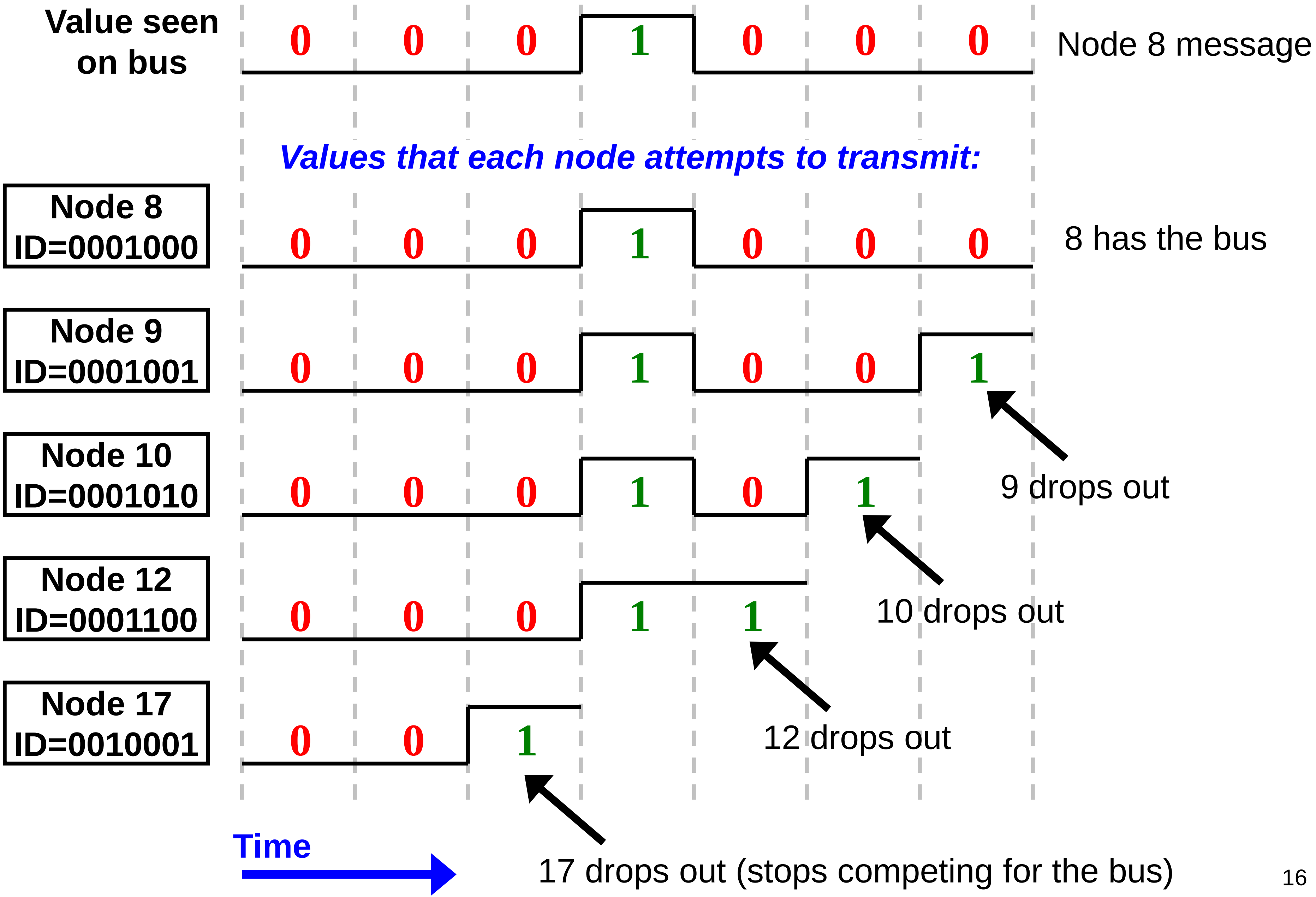
[Siemens]

CAN – Binary Countdown

- **Use bit dominance to decide who gets the bus**
  - Based on message header

- **Binary Countdown operation:**
  - Each node uses unique identifier for each message type
  - All nodes wishing to transmit compete for the channel by transmitting a binary signal based on their identification value
  - A node drops out the competition if it detects a dominant state while transmitting a passive state
    - If it transmits “1” and sees “0” on the bus, it knows it lost the arbitration
  - Thus, the node with the lowest identification value wins
  - Contention is resolved without consuming any extra bandwidth

Example: Binary Countdown



CAN - General Message Format

HEADER	DATA	ERROR DETECTION
19 or 39 bits	0 to 64 bits	15 bits

Header

- Application sets any desired value in 11-bit or 29-bit identifier
  - A few other fields in the header too (e.g., 4 bits for length of data field)
- Identifier determines global priority (which message gets on bus first?)
- Headers often chosen to reflect source, destination, and/or message type
  - But are entirely application-dependent
  - Broadcast messages – all receivers can receive every message by default

Data

- Application- or high-level-standard defined data fields
- CAN Data field size is variable (0 to 8 bytes, defined via length in header)

Error detection

- Detects corrupted data (uses a 15-bit CRC):
  - All 15-bit or shorter burst errors (groups of flipped bits clumped together)
  - All 5-bit errors regardless of where they occur (except for a design bug; see 18-649)

CAN in Real Time Systems

- **Why not just use Ethernet and TCP/IP?**
  - Collision avoidance uses random backoff and retry strategy
  - Worst case message delivery is infinite!
  - OK for best effort, but not real time
  - Message overhead is huge – what if you just want to send “switch X is ON”?
    - 18(min. Ethernet "header" fields) + 6(padding) + 20(IPv4) + 20(TCP) = 64 bytes (http://stackoverflow.com/questions/1846077/size-of-empty-udp-and-tcp-packet)

- **CAN message format is optimized for real time control**
  - Bus arbitration uses low overhead and is constant
  - Bit dominance enables global prioritization of messages
  - Can work efficiently on inexpensive 250K, 500K, or 1Mbit/sec networks

- **CAN allows real time schedulability**
  - Non-preemptive prioritized messages

- **Can predict worst case latency for message delivery**
  - As long as the system isn’t overscheduled
  - The math looks the same as the non-preemptive prioritized tasking equation!

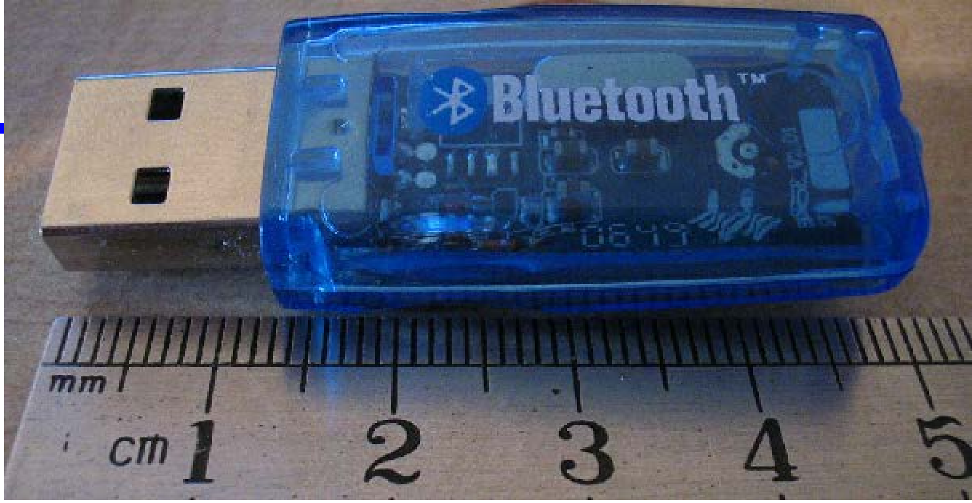
CAN Tradeoffs

- **Advantages**
  - Very low overhead (minimum size is only 44 bit message with 11 bit ID)
  - High throughput under light loads
  - Global prioritization possible, basically for free
  - Arbitration is part of the message - low overhead
- **Disadvantages**
  - Requires bit dominance
    - Restricts physical layer choices
  - Unfair access - node with a high priority can "hog" the network
    - Can be reduced in severity by using Message type instead of Node # for header
    - Can, in principle, use a bus guardian to limit duty cycle of each node
    - You have to pay attention to real time (e.g., Deadline Monotonic Scheduling)
  - Poor latency for low priority nodes
    - Starvation is possible
- **Typical performance:**
  100 bit messages @ 1 Mbit/sec ➔ 10,000 msgs/sec

Wireless Embedded Network: Bluetooth

Bluetooth

- **Open wireless protocol standard**
  - Published by Bluetooth Special Interest Group (SIG)
  - http://www.bluetooth.org



[Wikipedia]

- **Provides short range wireless communication**
  - 10 to 100 meter range
  - 1 Mbit/s (basic mode) up to 3 Mbit/s (enhanced data rate mode)
  - Primarily intended for voice and data transfer
  - Eliminates wires and cables between both stationary and mobile devices
  - Ad-hoc networks and synchronization of personal devices
  - Intended to be cheap (< $5 per node) to allow ubiquitous computing
    - This is still pretty expensive in embedded

Bluetooth - History and Applications

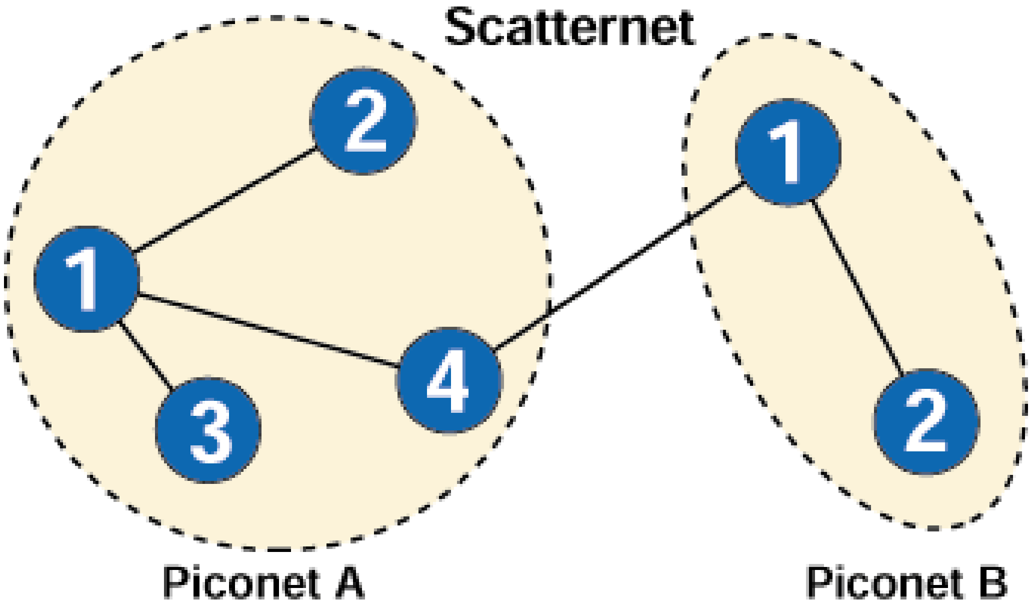
- **Harald Blåtand (Bluetooth in English)**
  - Viking and king of Denmark between 940 and 981
  - One of his skills was to make people talk to each other
  - During his rule Denmark and Norway were united

  … what they don't tell you is that his murder was arranged by his son, who coveted the throne

- **Goal: Bluetooth standard lets devices communicate and work together**
  - PCs and peripherals
  - Synchronization of PDAs and other personal devices
  - Cellphones and wireless headsets
  - Tethering (relay Internet connection from cell phone to laptop)

[Wikipedia]

Bluetooth - Network Topology



- **Ad-hoc network**
  - Any nodes that come within range can form small network called a “piconet”
    - Nodes can dynamically join and leave the ad-hoc network
    - Nodes can belong to multiple piconets
  - 1 master node and up to 7 slave nodes in a piconet
  - Elects a new master if master leaves network

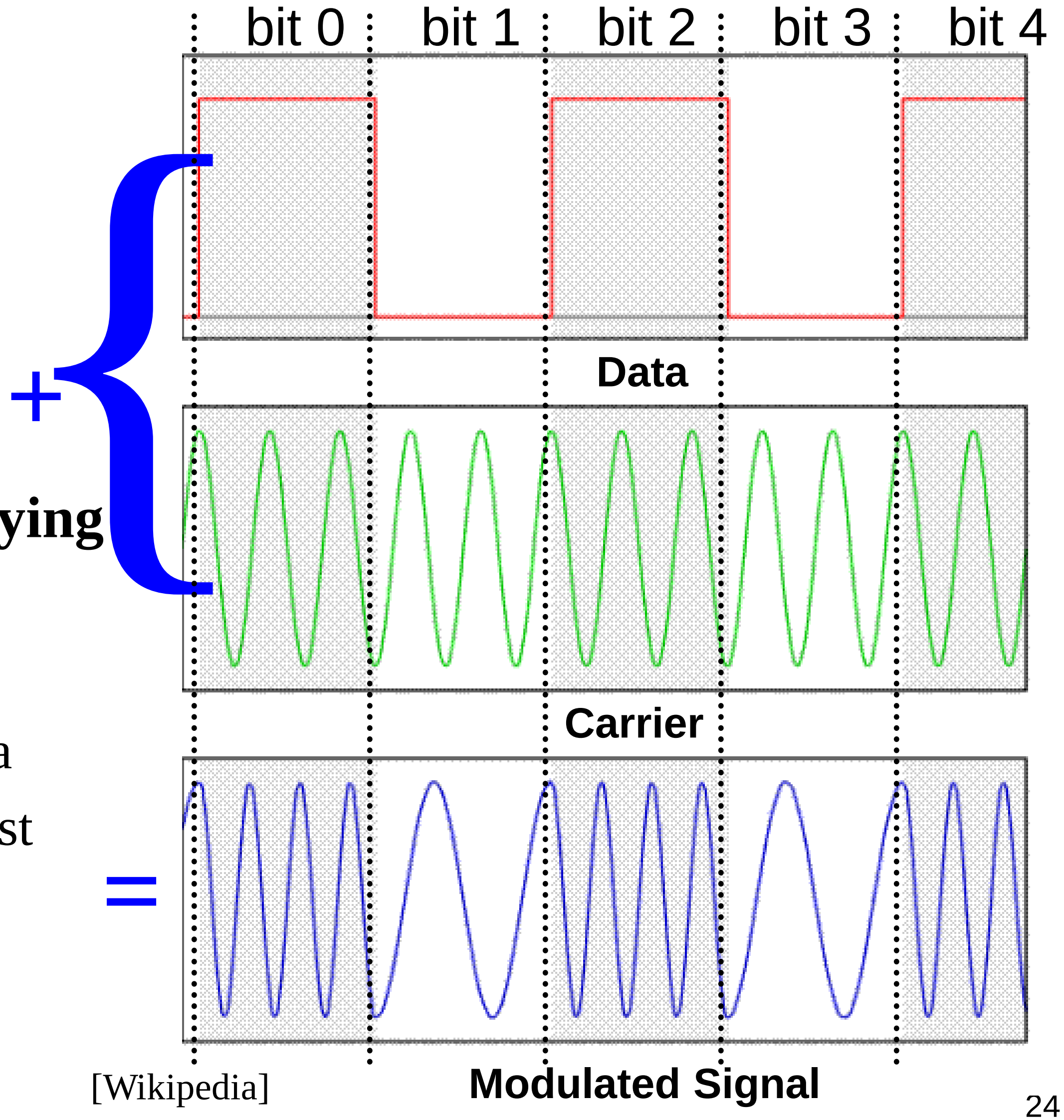
- **Scatternets**
  - If a node is part of two different piconets, can share information across them

Bluetooth - Physical Layer

- **Bluetooth uses RF as a wireless medium**
  - Uses the unlicensed 2.4 GHz Industrial-Scientific-Medical (ISM) radio frequency band
  - Why is ISM at 2.4 GHz?

- **Bits are sent using frequency modulation (FM)**

- **Gaussian Frequency Shift Keying**
  - Different frequencies to indicate a “1” or “0”
  - “Gaussian” just means it uses a smoothing filter on the data first to remove sharp edges



[Wikipedia]

Bluetooth - Media Access Strategy

- **Get collisions if two devices broadcast on same frequency at same time**
  - Need to prevent two nodes in piconet from causing collision
  - Need to prevent collisions caused by outside RF interference

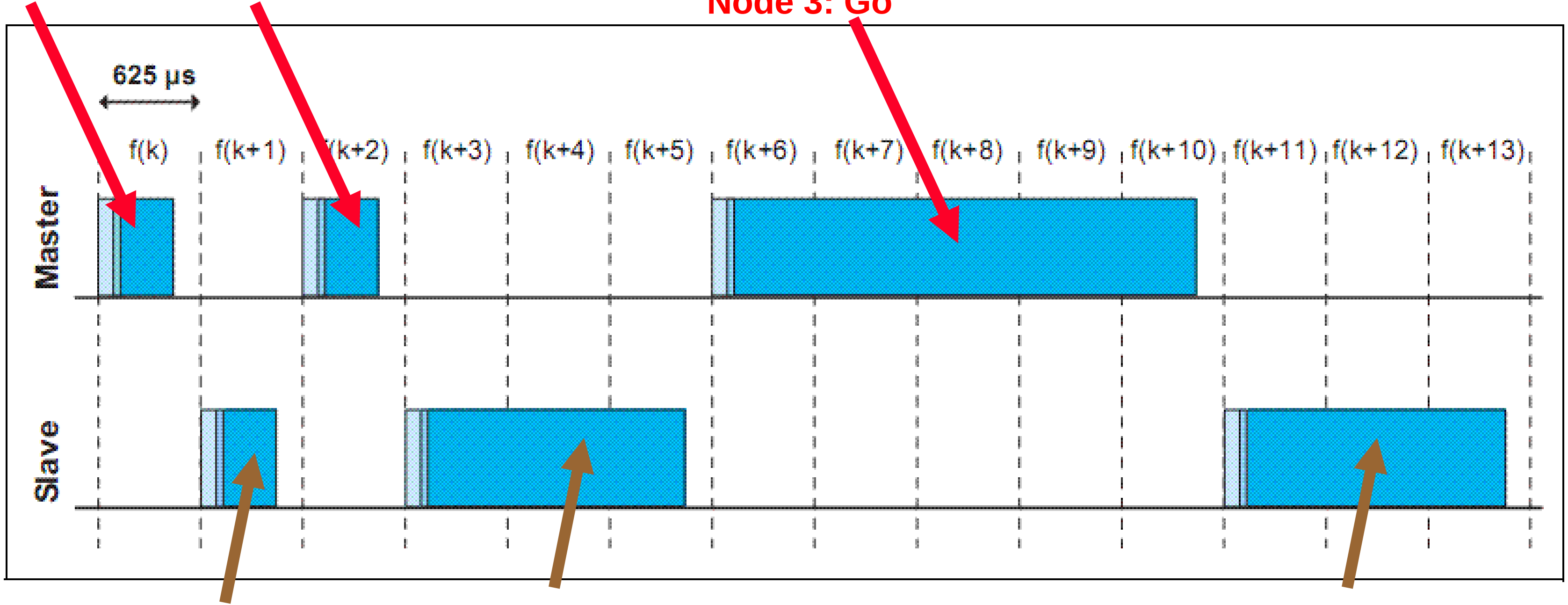
- **Bluetooth uses master-slave approach within the piconet**
  - Master controls which slave gets to broadcast next
  - Slaves wait to be given their turn

- **Frequency Hopping Spread Spectrum**
  - Constantly hop among psuedo-random set of frequencies
  - Reduce collisions caused by outside RF interference

Bluetooth - Master Slave

- **Master broadcasts every other message**
  - Sends data and declares who goes next; master relays data to other nodes
- **Time Division Duplex**
  - Time divided into slots
  - All slave nodes synchronize to master node’s clock (start of message)
  - Use a different frequency each time slot

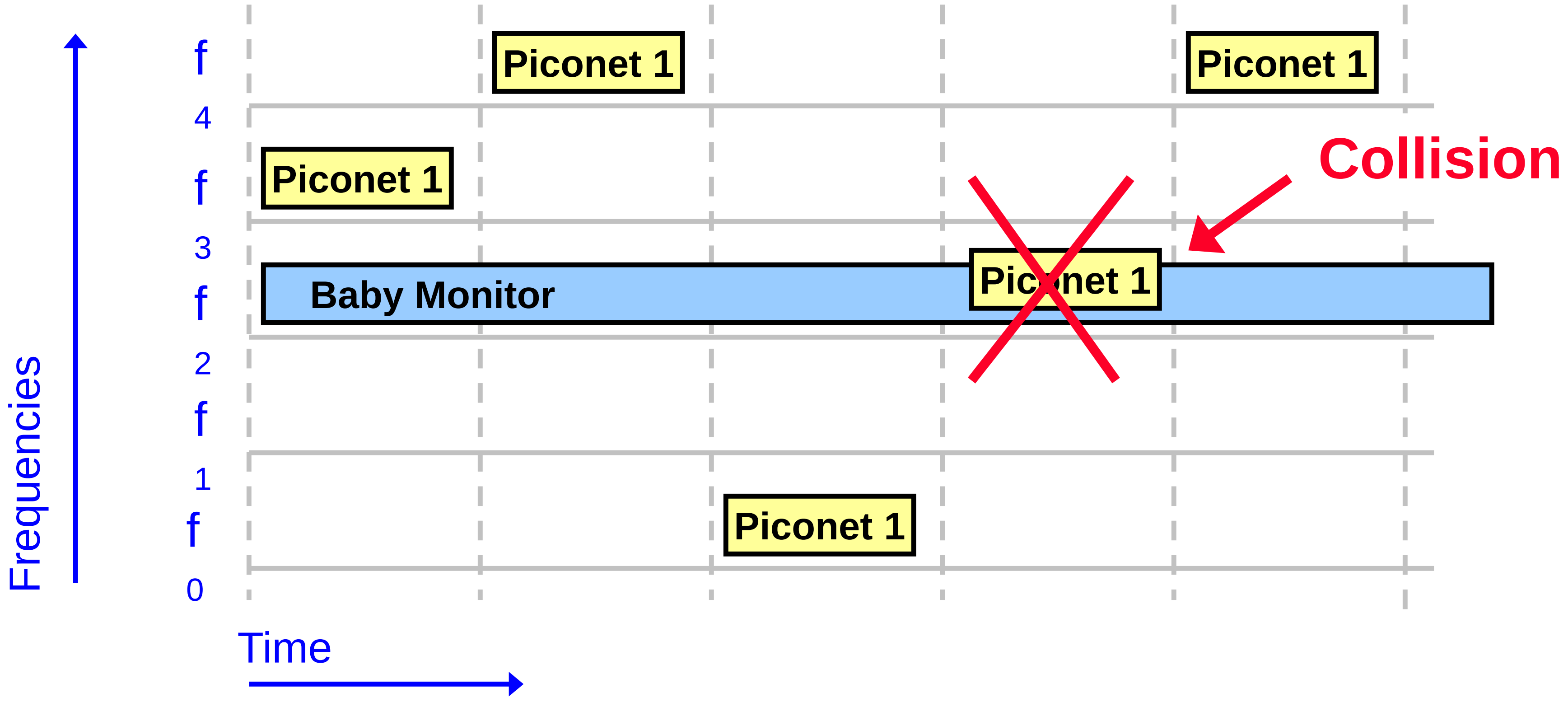
“Node 1: Go” “Node 2: Go” “Here is some data for node 3; Node 3: Go” [Bluetooth Core Spec 3.0]



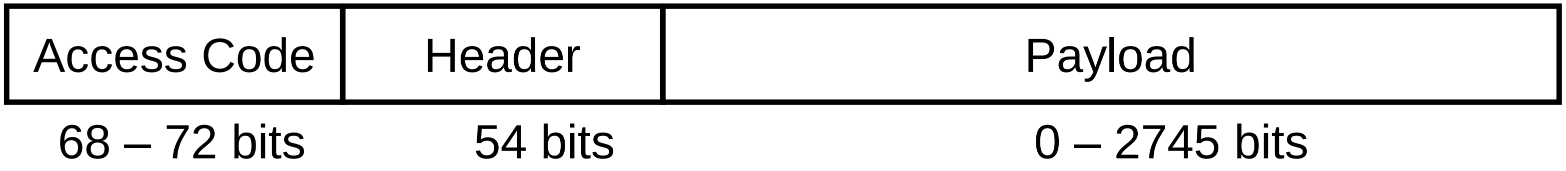
“Node 1 Response” “Node 2 Response” “Node 3 Response”

Bluetooth - Frequency Hopping Spread Spectrum

- **Master defines a psuedo-random series of frequencies**
  - Signal hops among frequencies to avoid interference
  - Tolerate the occasional collision; data lost but protocol recovers automatically
- **Other devices on the 2.4 GHz ISM band**
  - Garage door openers, baby monitors, microwave ovens
  - (2.4 GHz is a bit of a mess with different protocols using it different ways)



Bluetooth - General Message Format



Access Code	Header	Payload
68 – 72 bits	54 bits	0 – 2745 bits

Access Code

- Want to keep messages from two piconets separate if they happen to use same frequency pattern
- Master node defines an access code for its piconet
- Any message must first have a valid access code
- Includes some serious error detection/error correction

Header

- Contains destination or source address, packet type, flow control, and error checking code

Payload

- Application data, often voice or other media
- Includes 16-bit CRC to catch corrupted data

Bluetooth Tradeoffs

- **Advantages**
  - No wires!
  - Ad-hoc networking
    - Nodes can join and leave whenever they want
    - Overlapping/interacting/hierarchical networks (scatternets)
- **Disadvantages**
  - No way to distribute power (everything has to have a battery or power plug)
  - Geometry may introduce standing waves/fading
  - Interference from other RF emitters (EMI = ElectroMagnetic Interference)
  - Limited spectrum space
  - ***In general, unsuitable for use in critical applications that aren't fail-safe!***
- **Also, cost**
  - Bluetooth cheap, but still working on being super- "cheap"
  - Has to be able to beat a piece of copper and a plastic connector

CAN/Bluetooth Comparison

- **Both:**
  - Relatively efficient for relatively short messages
  - Optimized for single-hop network to connect physically close devices
  - Give predictable and bounded worst-case message delays
- **CAN:**
  - Wired network for critical control applications
  - Assumes fixed network configuration known at design time
  - Prioritized message sequence supports rate monotonic scheduling
  - Optimized for extremely short data payloads (e.g., 1-byte payloads)
- **Bluetooth:**
  - Wireless network for consumer electronics data exchange
  - Higher overhead, but longer maximum message size for bulk data transfer
  - Round-robin message sequencing for fairness

Other Wireless Alternatives

- **ZigBee – IEEE 802.15.4**
  - Lightweight, low-rate data protocol
  - Desgined as cheaper alternative than Bluetooth

- **Custom wireless protocols**
  - For example, lighting systems, remote automotive entry devices
  - For battery-powered equipment RF can take most of the energy budget
  - FCC regulates frequencies and how many bits you can send
    - Often only a few msec of data at a low data rate
    - Serious limitations on automatic devices (i.e., no “button press”)

- **Infrared – IrDA**
  - Infrared Data Association
    - http://en.wikipedia.org/wiki/Infrared_Data_Association
  - TV remotes … but also data transfers up to 1 Gbit/Sec with GigaIR
  - Good for indoors within same room; no RF interference from WiFi
    - Wii uses Bluetooth for data communications, IR for pointing

Review

- **Controller Area Network (CAN)**
  - Bit dominance and binary countdown

- **Bluetooth**
  - Wireless protocol
  - Frequency hopping to minimize effects of collisions

- **Comparisons between CAN and Bluetooth**