Embedded Systems and Ubiquitous Computing

Embedded Ecosystems

CSE 199

Fall 2017

Homework

For recitation this week, you must:

1. Log the embedded systems you use for 24 hours
2. Select two systems and write a paragraph about each
3. Turn in those paragraphs on ePortfolio
4. Read one (or maybe both) aloud in recitation

Read the assignment details on Piazza!

Embedded Ecosystems

Embedded systems are often part of a larger whole.

(Sometimes a much larger whole!)

We will look at some of these larger systems.

Types of Systems

What sorts of larger systems?

- Many systems in a single device
  - *E.g.*, an automobile
- Many systems in many devices
  - *E.g.*, a factory floor
- Regulatory frameworks
- Published standards

Not all systems are technical!

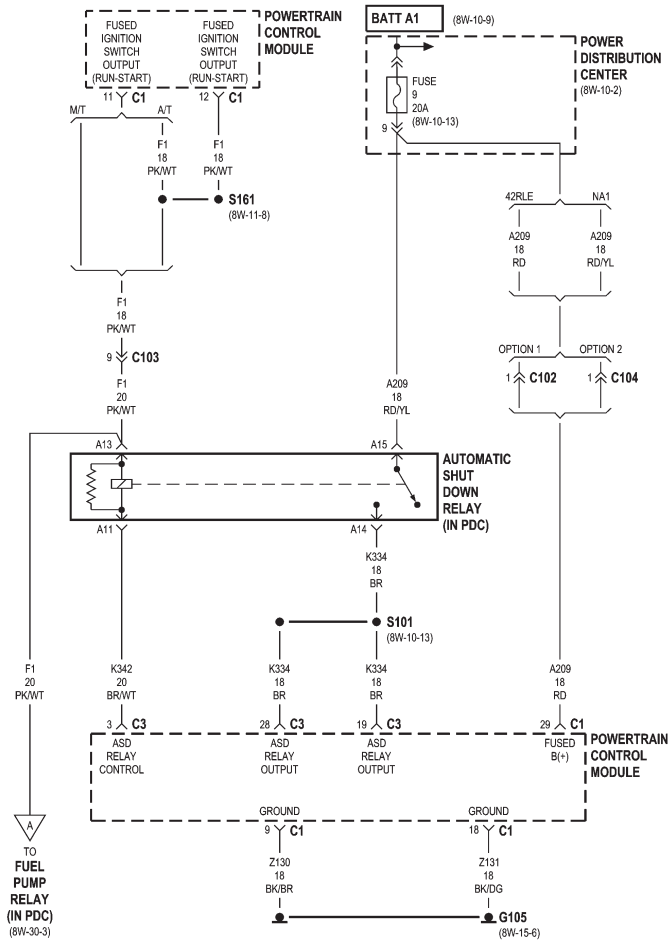
An Illustrative Example

Automotive systems provide a mature and complex example.

Modern cars contain numerous, interconnected systems.

These systems have safety implications and operate in a hostile environment.

Increasingly, they are also Internet-connected.



Automotive systems

From IEEE Spectrum, February 1, 2009:

[I]f you bought a premium-class automobile recently, "it probably contains close to 100 million lines of software code," says Manfred Broy, a professor of informatics at Technical University, Munich, and a leading expert on software in cars. All that software executes on 70 to 100 microprocessor-based electronic control units (ECUs) networked throughout the body of your car.

Even low-end cars now have 30 to 50 ECUs embedded in the body, doors, dash, roof, trunk, seats, and just about anywhere else the car's designers can think to put them.

Automotive systems

- **1977:** GM Olds Toronado has the first software ECU†
- **1978:** Cadillac Seville has optional "Trip Computer"†
- **1981:** *All* GM domestic passenger cars have software ECUs, ≈ 50 kLoC†
- **2005:** Electronics are 15% of vehicle cost (excluding final assembly)†
- **2015:** Ford GT has 10 MLoC, 28 processors
- **2016:** Ford F150 has 150 MLoC!

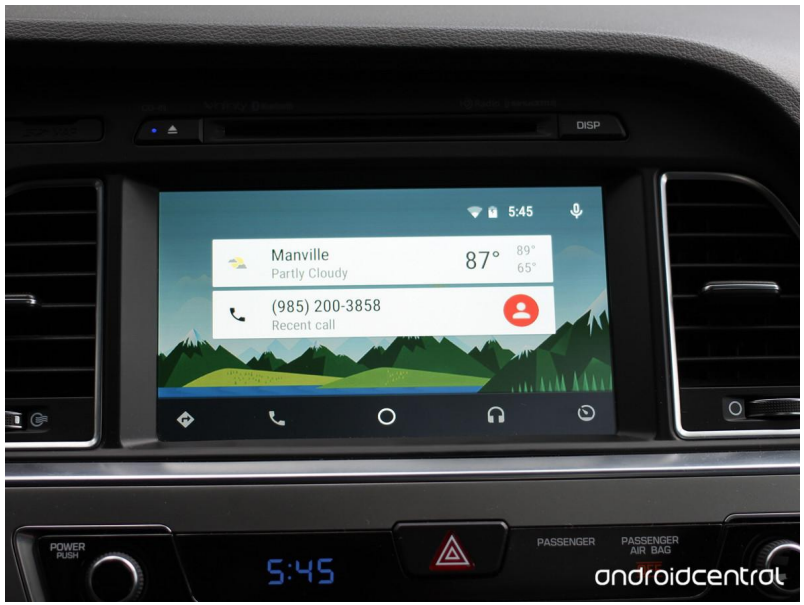
†IEEE Spectrum, February 1, 2009

150 Million Lines?

(Probably) the bulk of that code is infotainment/navigation.

A lot of the remaining code is diagnostics.

There's still a lot of *functional* code!



Typical Automotive Embedded Systems

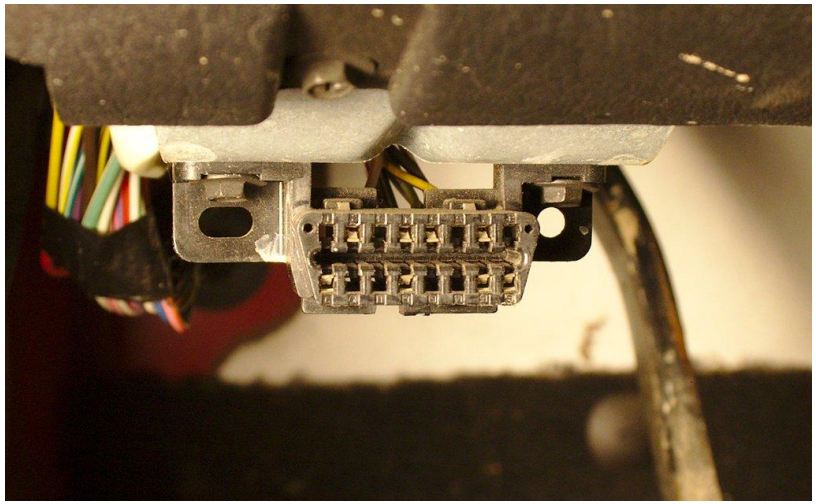
- Engine control
- Transmission
- Airbag computer
- Anti-lock brake system
- Tire pressure sensor system
- Door locks / windows / *etc.*
- RADAR collision or parking sensors
- Navigation
- Entertainment

Separation of Concerns

Dividing functionality into separate embedded systems has advantages.

- Reduced complexity of critical systems
- Robustness to failure
- Repairability
- Ease of update

Communication



Many automotive systems communicate with each other.

Some communicate with the outside world.

There are multiple networks in most automobiles.

There are also multiple network technologies!

Automotive Networks

CAN (Controller Area Network) bus:

- Electrically robust
- Moderate speed
- Multi-point, multi-master

LIN (Local Interconnect Network) bus:

- Electrically less robust
- Slow speed
- Multi-point, single master
- *Cheap*

OBD-II

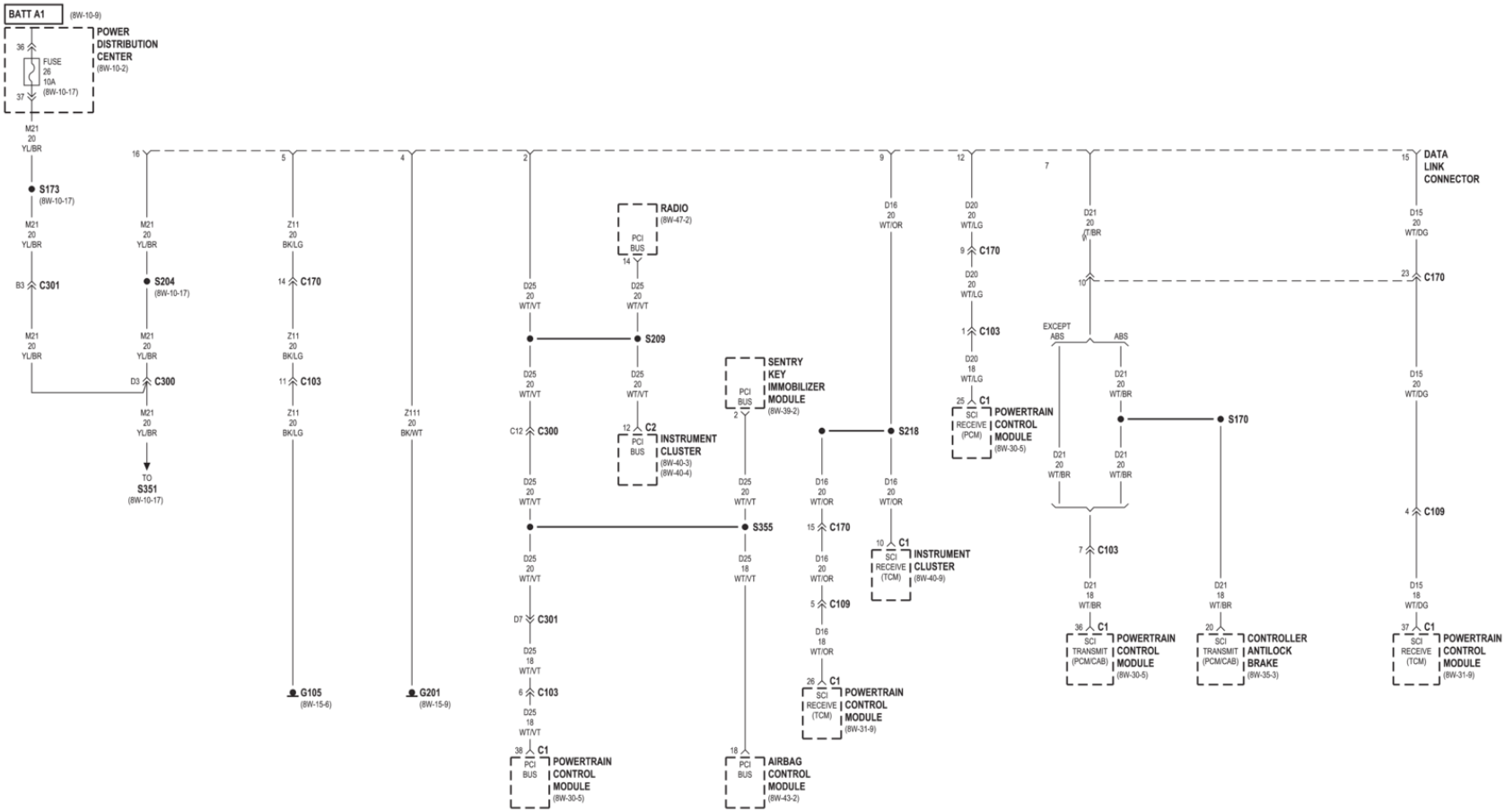
OBD-II is the second version of an On-Board Diagnostics protocol standard.

Mandatory for some or all cars sold in various areas.

- The State of California since 1996
- United States since 1996
- The EU (as EOBD) since 2001
- China since 2008

In the US, OBD-II uses CAN.

2005 Jeep TJ Bus Communications



Reliability

An automobile's environment is extremely harsh.

- Temperatures -40°C–125°C
- Prolonged, powerful vibrations
- Static charge buildup
- Wide power voltage fluctuations
- Powerful electrical noise

Special automotive versions of many parts are available.

Safety

Automobile safety is clearly important!

Many components are *safety-critical systems* (as discussed last lecture).

Independent safety-critical modules simplify design.

Mixed-criticality modules are also in use.

Safety Standards

Several standards (both mandatory and voluntary):

- IEC 61508:
  Functional Safety of Electrical / Electronic / Programmable Electronic Safety-related Systems
- MISRA-C:2004, MISRA-C:2012
  Guidelines for the use of the C language in critical systems (available online)
- ISO 26262
  Road vehicles — Functional safety

From *Safety in Automotive Software: an Overview of Current Practices*
P. Panaroni, G. Sartori, F. Fabbrini, M. Fusani, G. Lami

IEEE Computer Software and Applications 2008

Providing Safety

- Fault Avoidance
- Fault Removal
- Fault Detection
- Fault Isolation
- Fault Recovery

From *Safety in Automotive Software: an Overview of Current Practices*
P. Panaroni, G. Sartori, F. Fabbrini, M. Fusani, G. Lami

IEEE Computer Software and Applications 2008

Aviation Systems

Aviation is like automotive, but more so:

- More regulation
- Higher stakes
- More complex systems
- More standards

Aviation Challenges

An automobile might have one infotainment system.

An airliner might have four hundred.

Domestic flights have substantial network infrastructure.

Maintenance windows are small, modularity is key.

Aviation Networking

A typical airframe communicates with many networks!

- Radio location services (GPS, aviation-specific)
- ADS-B downlinks for position and status
- Voice communication
- In-flight entertainment
- In-flight customer Internet

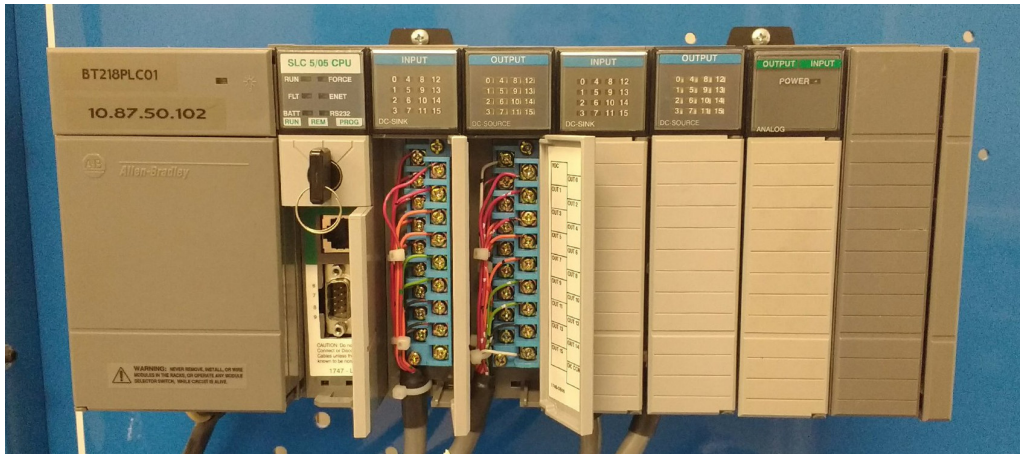
Industrial Systems

Industry uses enormous and complex systems.

Some industrial systems cover acres.

These large, complex systems may be globally unique

Special tools have been developed for industry.



Programmable Logic Controllers

PLCs are “generic” embedded systems devices.

They are extremely robust.

Modern PLCs are often Internet-connected.

Networks of PLCs or similar controllers may run:

- Paper machines
- Automotive assembly lines
- Toy injection molding machines

Example of Industrial Automation

This video is an advertisement, but shows some good industrial embedded systems.

Notice the variety of components working together:

- Conveyors
- Arms
- Sorters
- Controllers

Conclusions

An individual embedded system may be only a tiny cog.

Communicating systems can perform amazing tasks.

Many industries are reliant on embedded systems.

Regulation, standards, and best practices for embedded systems are legion.