Luke Jaffe
ID: 001182531
Microprocessor-Based Design
Due 02/05/16

Homework 1

Introduction to Embedded Systems

Problem 1.1: Definition

An embedded system is:

1. A system embedded into another system with an application specific purpose and constraints from external input. (lecture slides)

2. A computer system with dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. (Wikipedia)

3. A special-purpose system in which the computer is completely encapsulated by the device it controls. Unlike a general-purpose computer, such as a personal computer, an embedded system performs predefined tasks, usually with very specific requirements. (NCSU ECE department)

These definitions are similar, but diverge on some elements. Definitions 1 and 2 explicitly mention that an embedded system is a sub-system of some larger system. Definition 3 does not make this distinction, suggesting that an embedded system is just a computer inside some device. All definitions mention specific purpose/requirements. Definition 1 does not mention any computing element, though it is probably implied. Definition 1 mentions constraints from external input, whereas definition 2 mentions real-time computing constraints. A combination of definitions 1 and 2 may be best.

Problem 1.2: Market

Name and briefly outline 5 different markets of embedded systems. Provide four different examples for embedded systems in each of the markets.

1. Automotive: Modern automobiles contain many embedded devices, for controlling everything from steering to climate control to airbags. As vehicles become fully automated, the number of embedded system will continue to increase.

    i. Anti-Lock Braking System
    ii. Satellite Radio
    iii. Adaptive Cruise Control
    iv. Airbag Control

2. Personal: A range of personal devices using embedded systems have recently becomes available. This is a result of reduced cost and power consumption of embedded systems.

    i. Noise-Cancelling Headphones
    ii. E-readers
    iii. Music players
    iv. Fitness Devices (Jawbone, Fitbit, Basis)

3. Healthcare: Embedded systems are used for many healthcare applications, including a wide variety of scanning mechanisms. Other healthcare applications include fitness monitoring devices (jawbone, fitbit), cardio equipment (treadmills, ellipticals, bikes), and pacemakers.

i. MRI (Magnetic Resonance Imaging)
ii. Pacemaker
iii. Wearable fitness monitor
iv. Interactive treadmill

4. Industrial: The continued automation of industrial production is another big embedded system market. Many modern factory assembly lines which produce physical devices and products use embedded systems for nearly every stage. Mechanical arms are used to move and place parts, conveyor belts to transport them between stages, sensors to determine position/temperature of components.

i. Pick and place robot
ii. Welding robot
iii. Packaging/Palletizing machine
iv. SMD (surface mount technology)

5. Telecommunication: Telecommunication is founded upon embedded devices. Most devices that send, receive, or route data are embedded systems.

i. Network switches and routers
ii. Satellite transmitters
iii. WiFi card
iv. Radio devices

Problem 1.3: Challenges

1. Power consumption: Embedded systems are often contained within systems which have strict power constraints. If every embedded system in a car had high power consumption, the vehicles battery would be quickly depleted. Likewise, hand-held gadgets like cellphones should consume as little power as possible to reduce time between charges. Newer devices like smartwatches are possible partly because of advances in handling power consumption (either by reducing consumption of devices, or using more compact batteries).

2. Real-time: While not all embedded systems are real-time, most are. Embedded systems are almost always designed with a specific purpose/functionality, and that functionality usually has a time constraint. This can be a very short time constraint such as an airbag for a car, or very slow, as in the sampling rate of a spacecraft which must conserve power. In order to operate under hard real-time constraints, an embedded system must have a reliable computing platform, which can deterministically guarantee that a program will execute within a deadline. Real-time is further elaborated in problem 1.4.

3. Size: Since embedded systems are the “control” portion of many larger systems, they should take up relatively little space to leave room for components with mechanical design constraints. Again using cars as an example, with dozens of embedded systems in a given vehicle, they need to be small so they do not interfere with essential mechanical parts of the vehicle, or interfere with passengers. Weight may also be a concern for some devices containing embedded systems. For example, Amazon has recently been developing new drones for automated delivery. The embedded systems on these drones must be lightweight to save thrust for the actual packages.

4. Cost: Embedded systems may need to be more cost effective than what they are “replacing”. For example, in factory automation, if the cost of adding many embedded systems does not add enough efficiency to beat mechanical/human alternatives, then they may not be worth it. Also, some systems may contain many embedded systems, which must be cheap for the product as a whole to be marketable. Again using cars, there is a tradeoff between the functionality added by incorporating many embedded systems, and the cost of actually manufacturing/buying and integrating them.

Problem 1.4: Real-time

a. What is a definition of a real-time system?

Real-time computing describes hardware and software systems subject to a “real-time constraint”, for example operational deadlines from event to system response. Real-time programs must guarantee response within specified time constraints, often referred to as “deadlines”. A real-time system has been described as one which “controls an environment by receiving data, processing them, and returning the results sufficiently quickly to affect the environment at that time.”

b. Depending on the severity of consequences, real-time systems are classified into two categories. Name and explain both categories. Outline one example for each category.

Hard real-time systems: Temporal constraints must be met, otherwise defects could have a dramatic impact on human life, on the environment, on the system, etc. An example would be an airbag system in a car. If the airbag system does not currently measure a dangerous acceleration in time, and deploy the physical airbag, a passenger could be injured or die.

Soft real-time systems: Temporal constraints can sometimes be missed without dramatic impact. For example, a music player device. If a few bits are incorrect hear and there, the result will likely be inaudible to the user. Too many missing bits would cause low audio quality, and the system is clearly still real-time because music is fundamentally based on time.

c. What does real-time execution imply? Does it necessarily imply fast execution? If yes, why? If not, why not?

Real-time execution implies that an embedded system meets its “deadlines” in interacting with and responding to its environment. This does not necessarily imply fast execution. In fact, an embedded system should execute exactly as fast as it needs to, to meet its goals while maintaining other design constraints such as power and cost. For example, the turning mechanism in an oil tanker may have an extremely course sampling rate of its environment, since the tanker moves at such a slow rate.

d. Are all embedded systems real-time systems?

Not all embedded systems are real-time systems. For example, texting functionality on mobile phones is a non-real time application (on an embedded device). There is no guarantee when a text will be received after it has been sent. Sometimes text messages are lost and do not arrive at all. While modern mobile phones are venturing from embedded to smartphone territory, we can imagine a simple communication device that sends non-critical packets through a cellular network or over the internet as a similar example.

Problem 1.5: Reactive Systems

Definition: A reactive embedded system is an embedded system that has continual interaction with its environment, and responds to asynchronous stimuli.

Example: One example of a reactive embedded system is an airbag controller. It must continually interact with its environment (sample acceleration and temperature), then make a decision as to whether to deploy the airbag.

Requirements: In the US, regulations require deployment in crashes at least equivalent in deceleration to 14 mph. Nearly all airbags are also designed to deploy in the event of a vehicle fire, when temperatures reach 300-400 *F. The accelerometer would likely need to sample very frequently to ensure safety, maybe in the microsecond range. The temperature sensor would likely need to sample much less frequently, maybe in the millisecond range for situations where the vehicle catches fire very quickly/explodes.