# ECE Communication Networks 1
# Homework 1
Fall 2006

1. The following is a parity check matrix for a binary $[n, k]$ code $C$:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

   (a) Find $n$ and $k$.

   (b) Find the generator matrix for $C$.

   (c) List the codewords for $C$.

   (d) What is the code rate for $C$?

2. Let $C$ be the binary code $\{(0,0,1),(1,1,1),(1,0,0),(0,1,0)\}$.

   (a) Show that $C$ is not linear.

   (b) What is $d(C)$? (note: since $C$ is not linear, this cannot be found by calculating the minimum weight).

3. Show that $g(X) = 1 + X + X^2 + \cdots + X^{n-1}$ is the generating polynomial for the $[n, 1]$ repetition code.

4. In CRC, the bits that are appended for "error detection" are called the frame check sequence (FCS), or the CRC digits. Suppose that we have a frame with bits $(1,1,1,0,0,1,1,0)$ and suppose that the generator polynomial for our CRC code is $(1,1,0,0,1)$.

   (a) Calculate the FCS for this example. (Hint: it should only be 4 bits long)

   (b) Suppose that the received frame is $(1,1,1,0,0,1,1,0,0,1,1,0)$. Was there an error in transmission?

5. Bertsekas and Gallagher, Problem 2.16

6. Bertsekas and Gallagher, Problem 2.20

7. Bertsekas and Gallagher, Problem 2.32

8. Bertsekas and Gallagher, Problem 2.34

9. On the course website, please find the file TinyOS_Proj.zip, which contains three directories of files. The directories are as follows:

   - Forwarder/ - A very simple TinyOS program to use as a starting point. All the program does is listen for a message and then broadcast the same message once it hears anything.
   - system/ - has the full set of general system files for TinyOS. These are architecture independent, and hence are not specific a specific mote platform.
   - mica2/ - Has the files specifically for Mica2 motes. This is where most of the interesting radio stuff is.

   These files correspond to the communication (transmission and reception) functions in TinyOS. This is not the complete TinyOS (many files were removed in order to allow you to focus on relevant pieces). Your team should explore these files and attempt to reconstruct the flow and interconnection of the code/functions. It is not necessary (or expected) that you try to compile this code. Specifically, you should be able to explain:

- How and where is CRC checking performed?
- How and where is carrier sensing performed?
- What type of framing is used? How is the start and end detected?
- What operations are called during a transmit?
- What operations are called during a receive?

To help you get started, you might examine:

http://www.tinyos.net/tinyos-1.x/doc/stack.pdf - radio stack of the MICA mote. This is NOT the same as in the MICA 2, but there are some similarities (enough to decipher what you need).

http://www.tinyos.net/tinyos-1.x/doc/tutorial/index.html - TinyOS tutorial. Lesson 4 covers the radio, but doesn't discuss MICA2 specifically.