The Journey Inside^{SM}: Microprocessors
Student Handout: Bugs and Debugging

Bugs and Debugging

When an error is found in a set of instructions given to a computer, it is called a bug. The process of finding the error in a set of computer instructions is called debugging. A story from the history of computers explains how the term became firmly attached to computer technology.

Early computers used mechanical switches in their circuits rather than vacuum tubes or transistors. One day in 1945, when scientists were working with the Mark II—one of the earliest computers built—the computer just stopped. No one was able to say why the computer stopped and no one could get the computer to restart. The reason for the breakdown was finally discovered. A moth had flown into the computer and was caught in one of the thousands of mechanical switches. After the moth was removed, the computer worked just fine.

The moth was taped in the log book the scientists kept as they worked with the computer. The operator noted, "We have debugged the computer." The terminology became part of the standard vocabulary used in the field of computing. (Note: The original bug is on display at the Naval Weapons Museum in Dahlgren, Virginia.)

When you explored using the set of instructions for your robot in different situations, you may have found some problems. You may have bugs in your instructions.

Consider the set of instructions you wrote for your robot. When you moved the chair you had to write a different set of instructions. In fact, you need a different set of instructions for each distance from the wall.

Suppose your robot could "tell" it had reached the wall. Then, each time you use the instruction "walk" you can "ask" it if it is at the wall. Instructions in a computer language that ask questions are called conditionals. Now try adding conditionals to your robot language.

For the previous activity, your robot had only four words in its language.

Stand: Stand up   **Walk**: Take one step forward

Sit: Sit down   **Turn**: Turn 90 degrees to the right

Add the following words than can be used to test the robot's position. The words are defined in the robot language.

Wall?: Is the robot touching the wall (with its foot)?

Chair?: Is the robot touching the chair (with its legs)?

Activities

1. Write a set of instructions to move the robot from the chair to the wall and back again. Keep in mind that the instructions should work for a chair placed any distance from the wall.

2. Test and debug your instructions.

3. Move the chair to a different distance from the wall. Test and debug your set of instructions. Make sure that they will work for a chair placed at any distance from the wall.

4. This task requires your entire group. Complete the task as quickly as possible. You must consider everyone's safety, and all members of the team must participate. One group member will act as a timer.

   a. Choose a leader.

   b. Try the task (described below) once and time it.

   c. Talk about how you might make changes that will improve your time.

   d. Try the task again using your changes and again time it.

   e. Is there something else you can do to improve your time?

   f. Try the task again.

   g. After you are done, think about how you improved the process. What do you think improved your time the most?

Task: A ball must travel from the leader to each person in the group, one after the other. Each person touches the ball, one after the other, until the ball returns to the leader. Your group will be timed from the moment your leader transfers the ball to the first receiver until the time the ball is given back to the leader.