



SCIENCE	TECHNOLOGY	ENGINEERING	ART	MATHEMATICS
<div></div> <div><h1>STEAM LESSON</h1></div> <div></div>				
SCIENCE	TECHNOLOGY	ENGINEERING	ART	MATHEMATICS
*** TASK: COD 02 ***				
Humanbot II - Sandwich				
STEAM	✓	OUTLINE		
SCIENCE		<p>The students will learn to use the basic elements of a flowchart. They will begin to understand the way a computer "thinks" (or does not think), and they will see the challenges inherent in performing a task with one.</p> <p>The students will begin to recognize the differences between open-loop control and closed-loop control. (optional)</p> <p>http://www.bu.edu/gk12/tyler/lessons/PBJ/PBJ.htm</p>		
TECNOLOGY				
ENGINEERING				
ART				
MATHEMATICS				
OUTCOMES	✓	DIGITAL TECHNOLOGY		
ST2-2DP-T		selects and uses materials, tools and equipment to develop solutions for a need or opportunity		
ST2-3DP-T		defines problems, describes and follows algorithms to develop solutions		
ST2-11DI-T		describes how digital systems represent and transmit data		
ST3-2DP-T		plans and uses materials, tools & equipment to develop solutions for a need or opportunity		
ST3-3DP-T		defines problems, and designs, modifies and follows algorithms to develop solutions		
ST3-11DI-T		explains how digital systems represent data, connect together to form networks & transmit data		
COMMENTS				
YouTube Video: Making a sandwich https://www.youtube.com/watch?v=leBEFaVHIIE				

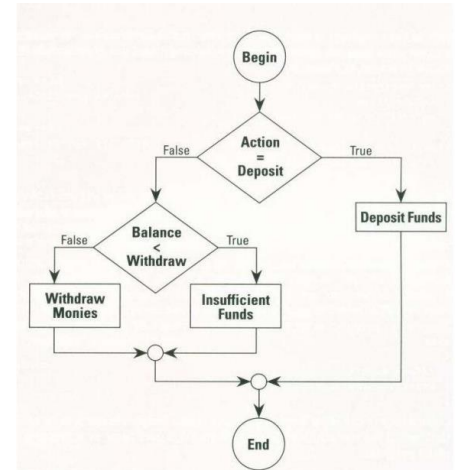
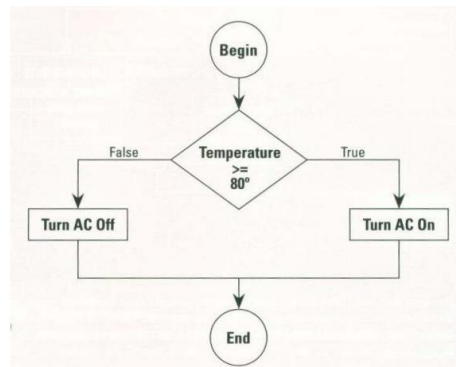
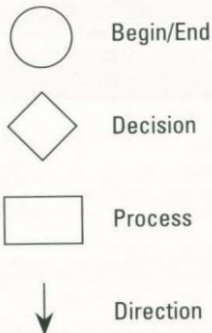
HUMANBOT II - EQUIPMENT

- Jar of Vegemite
- Plate
- Butter knife
- Two slices of bread



PRIOR KNOWLEDGE – COMMON FLOWCHARTS

Common Flowchart Symbols



OPENER

Sitting at the front with the sandwich materials laid out in front of you, introduce yourself to the class not as their teacher but instead as a highly functional cooking robot. Explain that when you entered the door, you left behind all of your experience and knowledge as a human being and that you're now set up to "think" and respond as a computer would. Their job as engineers will be to "program" you to make a vegemite sandwich.

There are many different levels at which you could present this challenge, depending on how much class time you have, what teaching points you would like to cover, etc., but we chose the following guidelines: The students may assume that 1) You have two arms which you can move up, down, left, and right. 2) You have wrists at which you can rotate your hands in either direction (clockwise and counter-clockwise). 3) Each hand has a "gripper" that you can open or close on command. They should assume nothing else!

DEVELOPMENT

At the top of the board, draw a circle with the word "begin", the standard start of any flow chart. Go around the room, asking each student to give you the next instruction necessary to make the sandwich. The basic goal for you is to simply respond the way you think a computer would. In other words, be very logical, literal, and disregard the normal thought processes that a human would make along the way. As each student gives you an instruction, ask them to come to the board and add their step(s) to the flow chart.

Again, you can really go as far with this as you choose, but here are some typical ideas the students will give, and how we responded, accompanied by teaching points.

HUMANBOT II - SCRIPT

DIALOGUE	TEACHING POINT
Student: <i>"Pick up the knife".</i> You: <i>"Does not compute. What is a knife?"</i>	<i>Remember, a computer has no concept of what a knife, spoon, vegemite, etc. are.</i>
Student: <i>"Move your arm forward".</i> You: Assuming you have an idea of where they're headed, move forward the opposite arm of what you think they'd like.	<i>The computer doesn't know what you're trying to accomplish, nor can it be made to understand, so the instructions must be specific. Which arm, the right or the left?</i>
Student: <i>"Move your left arm forward".</i> You: Move your left arm forward, and continue moving it forward (far past the vegemite jar or knife or whatever they're trying to get you to target) until they tell you to stop.	<i>Again, the computer doesn't know what you're trying to accomplish. It simply responds very literally to what it is told. After "turning a motor on", it will not turn the motor off until it is told. Apart from saying "stop" at the desired time, the student could also try saying, "move your left arm forward 10 cm".</i>
Assuming they get you to the point where you have one hand holding the knife inside the vegemite jar, getting vegemite on the knife can be quite challenging. Here's an example of the command that might happen...	
Student: <i>"Move your right hand (the one with the knife in it) around in a circle three times".</i> You: With your left hand at your side (since they'll likely be taking for granted how important both hands are in this step), move your right hand in a circle within the vegemite jar. When you do this without holding the jar with the other hand, you'll likely knock the jar over.	<i>You get the idea. They take for granted the common sense that using their other hand to hold the jar is necessary for this step. A computer won't think of this.</i>
After 20-30 minutes (and giving each student a chance to give some instructions), they'll likely have the main ideas. Explain that this has been practice at <i>open-loop control</i> , where the user has ability to control the process as it takes place (e.g. <i>"move right arm forward 10 cm"</i> , <i>"oh, move right arm forward 2 cm more"</i> , etc.). At some level, they'll hopefully come to appreciate how difficult controlling a computer/robot can be to perform a task as seemingly simple as making a vegemite sandwich.	
EXTENSION	
<p>Split them up into groups and challenge them to write up a flow chart for this process as a <i>closed-loop control system</i>. Ask them to imagine setting up their robot in front of the sandwich materials, pressing the "go" button on the robot's back, and then heading for the couch to watch TV. After all, who wants a robot whose hand needs to be held every step of the way?</p> <p>The robot has to make the sandwich on its own without any input from the students, meaning that the flow chart for the robot must include logical criteria along the way to help the robot "think" through the process the way they would. For example: Does the loaf contain at least two pieces of bread? If so, take two pieces. If not, go to the cupboard and open a new loaf.</p>	

HUMANBOT II - CLOSURE

Ask the students what they would take away from this experience for designing an automated (computer controlled) process in the future. There are many points that can be made in going through this activity, but here are a few major ones that they'll hopefully catch onto:

- **Computers (or highly functional cooking robots that think like computers) have no concept of common sense the way human beings do.**
 - If told to move an arm, they will continue moving an arm until they are told to stop. Even if that arm jams into a wall, a computer will continue trying to move it if not told to stop, maybe even burning out a motor in the process.
- **Computers need very specific, quantitative instructions.**
 - "Move your hand to the peanut butter jar" doesn't cut it for a computer. Instead, it must be "move your hand 4 inches forward", "rotate your hand 30 degrees clockwise", etc

Here are a few points more specific to talking about *closed-loop control*, should you choose to do so:

- **A good "program" or process must account for all possibilities that can occur.**
 - Example: It's possible that their highly functional cooking robot could be sent into the kitchen to make a peanut butter and jelly sandwich where someone left the peanut butter jar open. Likely their flowchart will include instructions to open the peanut butter jar. Suppose these include a command such as "rotate your hand until the cap becomes loose". How would the robot respond if the cap were already off? It might cause an error, or even worse, the robot might rotate it's hand indefinitely, never having the appropriate feedback to stop (aka an "infinite loop"). A good programmer/engineer will recognize this potential glitch and include a question in the flow chart: *Is the peanut butter jar open? If not, ... (open the jar). If so, ... (grab the knife, and so on).*
- **Feedback in closed-loop control must be quantitative and logical.**
 - A computer has no concept of what a "good" peanut butter and jelly sandwich is. The program must provide it with some quantitative measures to help along the way.
 - As an example, how should the computer know when it has put enough peanut butter on the bread? What does "enough" mean to the robot, who has never tasted a peanut butter and jelly sandwich? Maybe the robot could weigh the bread with peanut butter. If the slice weighs ____ grams or more, the robot should move onto the jelly. If not, the robot should apply more peanut butter. Or maybe the robot could measure the % of the bread's surface area covered by peanut butter and use this as a decision in the flow chart of how to proceed next.

EVALUATION

After the students have finished their flow charts for the closed-loop control, ask them to team up with another group. Try having members from the opposite group act out the flow chart the way they think a robot would.