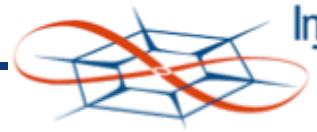
Flowcharts

Gary J. Minden
January 31, 2017

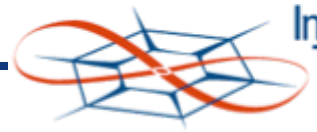


Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

Program Design

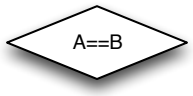
- Before writing code, we'll design an initialization subroutine and an execute subroutine
- We'll use a technique call Flow Charts that is a graphical representation of the program
- Flow Charts use a set of symbols to represent different actions of a program
- You can use Microsoft Visio on the EECS Windows computers to generate Flow Charts



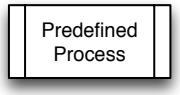
Flow Charts -- Basic Symbols

- The entry and exit for a process is denoted with a rounded rectangle

- Sequential code is indicated by a rectangle



- A conditional test is indicated by a diamond



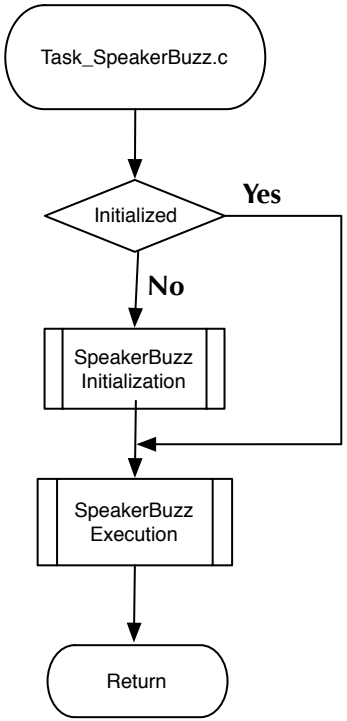
- Invoking a subroutine is denoted by a rectangle with two vertical lines.

- An off-page connector is denoted by a circle.

Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

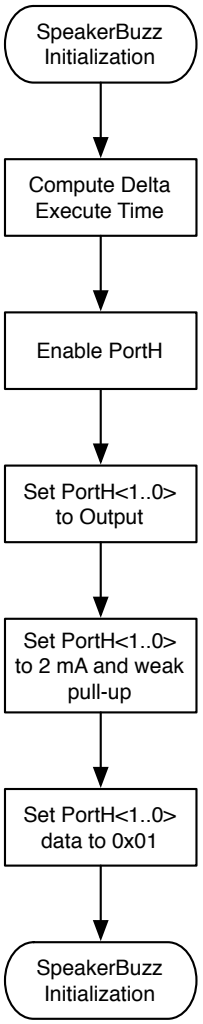
SpeakerBuzz Task



Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

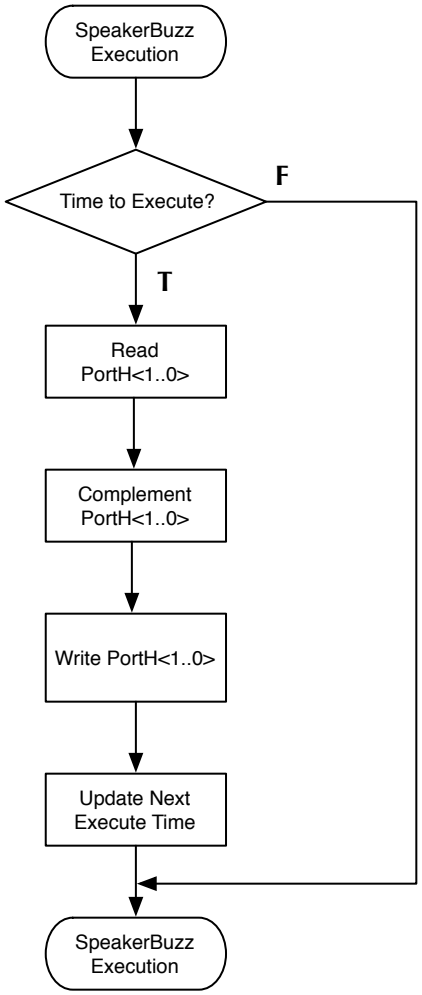
SpeakerBuzz Initialization



Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

SpeakerBuzz Execute

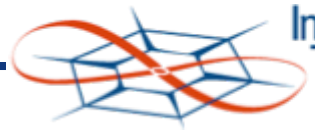


Information and Telecommunication Technology Center

THE UNIVERSITY OF KANSAS

SpeakerBuzz -- Next Steps

- Flow Charts help us visualize the organization of our programs
- The first set of Flow Charts describe what we want the program to do, not how to do it
  - The "What" becomes our program comments
- We refine our Flow Charts to specify how to accomplish our program goals
  - We use the subroutines in TI DriverLib to implement our program



Pseudocode

- A second approach is to write the outline of a program in a "Pseudo-language"
- The focus is on what to do, not the how or syntax
- Basic constructs:

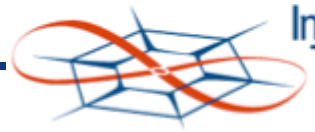
FUNCTION is description of a single task in design

SEQUENCE is a linear progression where one task is performed sequentially after another.
WHILE is a loop (repetition) with a simple conditional test at its beginning.
IF-THEN-ELSE is a decision (selection) in which a choice is made between two alternative courses of action.

REPEAT-UNTIL is a loop with a simple conditional test at the bottom.
CASE is a multiway branch (decision) based on the value of an expression.
CASE is a generalization of IF-THEN-ELSE.
FOR is a "counting" loop.

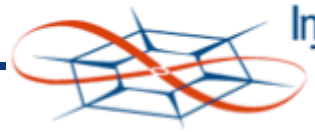
SpeakerBuzz -- Task

```
FUNCTION Task_SpeakerBuzz {

      IF ( <Task_SpeakerBuzz not initialized> ) {
         <SpeakerBuzz_Init>
      }

      <SpeakerBuzz_Execute>
}
```



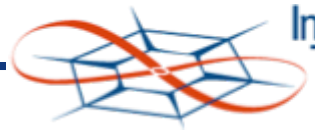
SpeakerBuzz -- Initialization

```
FUNCTION SpeakerBuzz_Init {
      <Compute delta execution interval>
      <Enable PortH>
      <Set PortH(1..0) to output>
      <Set PortH(1..0) to 2 mA output drive>
      <Write PortH[1..0] <= "01">
}
```



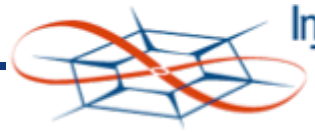
SpeakerBuzz -- Execution

```
FUNCTION SpeakerBuzz_Execute {
      IF (<Time to execute>) {
          <Read PortH[1..0] to Temp>
          <Complement Temp[1..0]>
          <Write PortH[1..0]>
          <Update next execution time using delta interval>
      }
}
```



Development of Pseudocode

- Initial Pseudocode can be vague
  - What does <Time to execute> mean?
  - What should the "delta interval" be?
  - What does it mean to <Enable PortH>
- Refine Pseudocode
  - Recognize <Time to execute> requires a CurrentTime and and ExecuteTime
    - These become state variables in your program
  - Recognize the "delta interval" is related to the sound frequency you want to generate
    - This is determined by the physical system
  - Recognize that <Enable PortH> is a hardware related function
- Refine your Pseudocode **before** writing your program



THE UNIVERSITY OF KANSAS

Additional Resources

- Creately flowchart tutorial: http://creately.com/blog/diagrams/flowchart-guide-flowchart-tutorial/
- Pseudocode: http://www.ittc.ku.edu/~gminden/Embedded_Systems/PDFs/Pseudocode_Standard.html
- Microsoft Visio: http://technology.ku.edu/office
- A Flow Chart "Cheat Sheet" is at: http://www.ittc.ku.edu/~gminden/Embedded_Systems/PDFs/flow-chart-symbols.png
- IBM's Flow Chart Techniques is at: http://www.ittc.ku.edu/~gminden/Embedded_Systems/PDFs/IBM-FlowchartingTechniques-GC20-8152-1.pdf

Actions

- Read the Creately tutorial at: http://creately.com/blog/diagrams/flowchart-guide-flowchart-tutorial/
- Read the Pseudocode_Standard: http://www.ittc.ku.edu/~gminden/Embedded_Systems/PDFs/Pseudocode_Standard.html
- Be prepared to apply these techniques in class