CS 91.204 Computing IV: Advanced OO Programming and Software Tools

Scribbler Programming

Prof. Li Xu
Dept. of Computer Science
UMass Lowell
Spring 2009

Scribbler Hardware

- Processor: 8-bit PIC microprocessor
- Clock: 20MHz
- RAM: 32 bytes
- EEPROM: 2K bytes
- I/O pins: 16
- PBASIC commands: 42

Scribbler Hardware

- Sensors
  - light sensors: 3, pin 0, 1, 2
  - line sensors: 2, pin 4, 5
  - stall sensors: 1, pin 7
  - obstacle sensors: 3, pin 6, 14, 15
- Motors
  - DC motors: 2, pin 12, 13
- Speaker: 1, pin 11
- LED: 3, pin 8, 9, 10

Scribbler Programming

- BASICStamp microcontroller execute compiled PBASIC code (tokens)
- PBASIC: similar to BASIC, with low-level commands to control I/O pins
- Refer to BasicStamp manual PDF for PBASIC commands

Hello, world

- hello.bs2

```
' {$STAMP BS2}
' {$PBASIC 2.5}
LED_left PIN   10
LED_center PIN 9
LED_right PIN 8
Main:
    DEBUG CLS, “Hello, world”, CR
    LED_left = 1
    LED_center = 1
    LED_right = 1
END
```

PBASIC Data Types

- No stack on Stamp microcontroller
- Program variables must be in RAM, bit, 4-bit, byte and word (2-byte) addressable
- ROM for array data and program binary code
  - restriction on ROM array: array value must be loaded to RAM variable before use
- Programmer will lay out RAM variables – 26 bytes available: be careful, you may run out of RAM!

PBASIC Data Types

- Data type
  - bit, nib, byte, word, 1, 4, 8, 16 bits
  - variables in RAM - 16 word (32 bytes) total
  - 3 reserved words: INS, OUTS, DIRS for 16 I/O pin control
  - DIRS set pin input/output: 0 for input, 1 for output: eg. set all input DIRS=%0000000000000000
  - INx, OUTx bit variables for each pin
  - Other RAM space for user variables

Variables

- RAM variable:
  foo VAR size (BIT, NIB, BYTE, WORD)
- Array variable:
  foo VAR size (n)
  eg, myList VAR BYTE (10)
- To use array:
  x = myList (3)

Numbers

- PBASIC uses 2's complement integer
  - decimal: 99
  - binary constant: %0010
  - hex number: $00FF
- Define constant:
  eg. foo CON "A"
  bar CON $80

PBASIC Expressions

- Arithmetic, logic operators
  - +, -, *, /, // for mod,
  - MIN, MAX, bit-wise &, |, ^
  - =, <>, <, <=, >, >=
  - AND, OR, NOT, XOR

PBASIC Stmts

- Control flow
  - assignment
  - if then else
  - goto
  - gosub
  - return
  - do … loop
  - for … next
- I/O
  - input, output: set pin as input, output; eg. output 7; out7=0
  - low, high: set pin to low, high voltage; eg. high 7
  - toggle: toggle pin level
  - pulsin, pulsout: send pulse to pin, control motor
  - freqout: send signal to pin
- Refer to the BASIC stamp reference manual

Example

```
' {$STAMP BS2}
' {$PBASIC 2.5}
square          VAR     Byte            ' FOR/NEXT counter

Setup:
  square = 1
Main:
  FOR square = 1 TO 250 STEP 2    ' show squares up to 250
    DEBUG DEC ? square                 ' display on screen
  NEXT                                 ' loop until square > 250
  END
```

I/O Pin

- Pin variables:
  - Example: mypin PIN 1
  - set pin variable to 0 or 1 will set pin's output voltage level to low or high, thus control I/O device, eg. sensors, motors

Subroutines

- Similar to functions, but:
  - no arguments
  - no local variables
  - no return value
- This is because there is no hardware stack

Host Communication

- Serial port connection to host PC
- DEBUG send bytes to PC
- DEBUGIN read bytes from PC
- Can use to print text and value and read data