

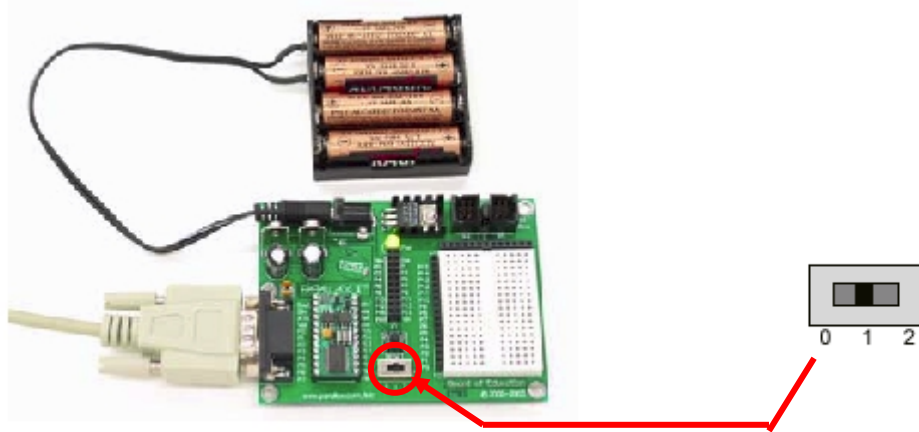
Team Assignment #2 Communicating with the BOE-BOT

Reading Assignment:

- 1) Refer to the following sections in Robotics Version 2.2 (instructor's webpage or online at www.parallax.com)
 - Chapter 1, Activity 3 – Setting Up the Hardware and Testing the System (pp 13 - 22)
 - Chapter 1, Activity 4 – Your First Program (pp 22 - 30)
 - Chapter 2, Activity 2 – Tracking Time and Repeating Actions With a Circuit (pp 45 – 58)
- 2) Refer to various sections of BASIC Stamp Syntax and Reference Manual, Version 2.1 for information of PBASIC commands (instructor handout or available online at www.parallax.com)
- 3) Lecture #14 – The BOE-BOT

Team Assignment:

- 1) Setting Up the Hardware and Testing the System
 - Obtain a BOE-BOT from the instructor. Record the number on the BOE-BOT box. Include this number on the title page of all reports involving the BOE-BOT. Be sure that you use the same BOE-BOT each class.
 - ***Use care when handling the BOE-BOT!***
 - Connect the serial cable from the computer to the BOE-BOT.
 - Connect the battery pack to the BOE-BOT (be sure to disconnect this at the end of the class.)
 - Turn the BOE on (move the slide switch to position 1 or 2).



- Launch the BASIC Stamp Editor.
- Select **Run – Identify** from the main menu in the BASIC Stamp Editor. A window like the one shown below should indicate that a BASIC Stamp 2 was found on COM1 on your computer. Check with the instructor if this does not work properly.

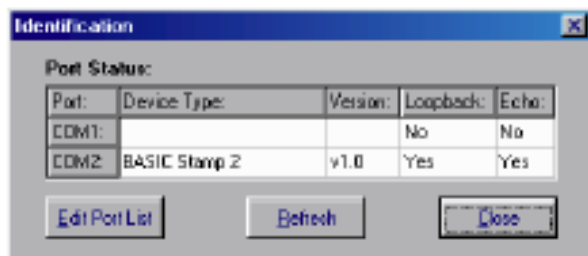


Figure 1-33
Identification
Window

*Example: BASIC
Stamp 2 found on
COM2.*

2) Your First ProgramA) Program 2A – Sending a message from the BOE-BOT to the computer screen

- Launch the BASIC Stamp Editor.
- Insert directives in your program:
 - Specifying a **BASIC 2** Stamp
 - Specifying **PBASIC 2.5** as the programming language
 - Specifying **COM1** as the communications port
- Write a program similar to the one shown on p. 25 of Robotics, Version 2.2 (shown below) except make the following changes:

```
' Robotics with the Boe-Bot - HelloBoeBot.bs2
' BASIC Stamp sends a text message to your PC/laptop.

' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "Hello, this is a message from your Boe-Bot."

END
```

- Include comments specifying:
 - course number
 - team assignment number
 - BOE-BOT number
 - your team number
 - each team member's name (only if present)
 - the program number (Program 2A, etc)
 - the filename (pick an appropriate filename).
- Change the greeting to "Hello Group N, this is a message from your BOE-BOT", where N is your group number.
- Run the program and verify that it works properly.
- Demonstrate proper operation of the program to the instructor.
- Save and print the program.
- Print the output of the program from the DEBUG screen..

B) Program 2B – Sending the result of a calculation from the BOE-BOT to the computer screen

- Make additions to Program 2A above like those on p. 29 of Robotics, Version 2.2 (shown below)
 - Run the program and verify that it works properly.
 - Demonstrate proper operation of the program to the instructor.
 - Save and print the program.
 - Print the output of the program.

```
' Robotics with the Boe-Bot - HelloBoeBotYourTurn.bs2
' BASIC Stamp does simple math, and sends the results
' to the Debug Terminal.

' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "Hello, this is a message from your Boe-Bot."
DEBUG CR, "What's 7 X 11?"
DEBUG CR, "The answer is: "
DEBUG DEC 7 * 11

END
```

3) Tracking Time and Repeating Actions With a CircuitA) Program 3A – Printing a message from the BOE-BOT once per second to the computer screen

- Write a program like the one on p. 43 of Robotics, Version 2.2 (shown below) except:

```
' Robotics with the Boe-Bot - TimedMessages.bs2
' Show how the PAUSE command can be used to display messages at human speeds.

' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "Start timer..."

PAUSE 1000
DEBUG CR, "One second elapsed..."

PAUSE 2000
DEBUG CR, "Three seconds elapsed..."

DEBUG CR, "Done."

END
```

- Include initial comments as in previous programs.
- Add additional comments to explain different PBASIC commands that are used
- Modify the program so that it displays:

```
Start timer ...
One second elapsed ...
Two seconds elapsed ...
.
.
Ten seconds elapsed ...
Done
```

- Run the program and verify that it works properly.
- Demonstrate proper operation of the program to the instructor.
- Save and print the program.
- Print the output of the program.

B) Program 3B – Turning ON and OFF an LED

- Write a program like the one on p. 50 of Robotics, Version 2.2 (shown below) except:

```
' Robotics with the Boe-Bot - HighLowLed.bs2
' Turn the LED connected to P13 on/off once every second.

' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "The LED connected to Pin 13 is blinking!"

DO
  HIGH 13
  PAUSE 500
  LOW 13
  PAUSE 500
LOOP
```

- Include initial comments as in previous programs.
- Add additional comments to explain different PBASIC commands that are used
- Use the output pin corresponding to your group number.
- Run the program and verify that it works properly.
- Demonstrate proper operation of the program to the instructor.
- Save and print the program.

C) Program 3C – Turning ON and OFF two LEDs (alternating)

- Write a program like the second one on p. 53 of Robotics, Version 2.2 (shown below) except:

```
DO
  HIGH 12
  HIGH 13
  PAUSE 500
  LOW 12
  LOW 13
  PAUSE 500
LOOP
```

- Include initial comments as in previous programs.
 - Add additional comments to explain different PBASIC commands that are used
 - Use the output pin corresponding to your group number and the next higher pin.
 - Modify the program so that the first LED is on for two seconds while the second LED is off and then the second LED is on for two seconds while the first is off (i.e., the LED's alternate blinking indefinitely).
 - Run the program and verify that it works properly.
 - Demonstrate proper operation of the program to the instructor.
 - Save and print the program.
- D) Program 3D – Simulating a traffic light using red, yellow, and green LEDs
- Write a program to control green, yellow, and red LEDs like a traffic signal. The green light should come on for 10 seconds, followed by the yellow for 1 second, and then the red for 7 seconds. The program should loop indefinitely.
 - Include initial comments as in previous programs.
 - Add additional comments to explain different PBASIC commands that are used
 - Use any three output pins.
 - Run the program and verify that it works properly.
 - Demonstrate proper operation of the program to the instructor.
 - Save and print the program.
- E) Program 3E – Turning ON and OFF an LED using the PULSOUT command

- Write a program like the second one on p. 55 of Robotics, Version 2.2 (shown below) except:

```
' Robotics with the Boe-Bot - PulseP13Led.bs2
' Send a 0.13 second pulse to the LED circuit connected to P13 every 2 s

' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "Program Running!"

DO
  PULSOUT 13, 65000
  PAUSE 2000
LOOP
```

- Include initial comments as in previous programs.
- Add additional comments to explain different PBASIC commands that are used
- Use the output corresponding to your group number.
- The program should turn on the LED for 0.1 seconds and turn it off for N seconds, where N is your group number. The program should loop indefinitely.
- Run the program and verify that it works properly.
- Demonstrate proper operation of the program to the instructor.
- Save and print the program.

4) **Report**

Organize your results into a report and submit a single typed report for the group to the instructor by the assigned due date. The report should consist of:

- A) A **title page** as shown below.
- B) Printouts for each of the programs (2A, 2B, 3A, 3B, 3C, 3D, and 3E).

EGR 120
Introduction to Engineering
Team Assignment #2
Date

Group #N (your group number)

Attendance & Participation Record:

(list all team members and all dates when teams worked together in class on this assignment and check boxes to mark attendance)

Team Member	Date 1	Date 2
John Doe	✓	✓
etc		

Demonstration of Programs

Program	Successfully Demonstrated
2A	✓
2B	✓
3A	✓
3B	✓
3C	✓
3D	✓
3E	✓