معـــهـــد التكنولوجـيا التطبيقيـــة

INSTITUTE OF APPLIED TECHNOLOGY

Micro-Controllers

Module 2: Outputs Control and Inputs Monitoring

PREPARED BY

IAT Curriculum Unit

August 2008

Module 2: Outputs Control and Inputs Monitoring

Module Objectives

1. Learn how to configure a Micro-Controller pin
2. Learn how to control an LED
3. Introduce Numbring systems (Decimal and Binary Systms)
4. Introduce variables and Loop commands
5. Learn how to monitor an input
6. Learn about IF … THEN commands

Module Contents

I/O Pin Configuration

Where are you now?

In module 1 you understood what a Micro-Controller is, and what are its applications. You also got familiarized with the hardware that you will use through out this course (BOE). Even more, you have written your first program and tested it successfully.

What's next?

In this module you will learn how to configure your I/O ports.
Also you will build your first Micro-controlled electric circuits and implement the electrical basics you have learnt so far. A typical circuit would look like the circuit illustrated in figure 2.1.

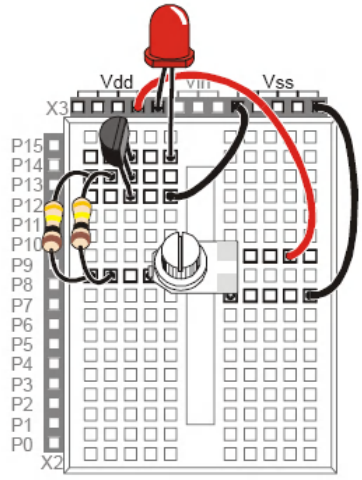


Fig 2.1: Typical electric circuit in BOE

Basic Stamp I/O pins

Basic Stamp 2 has 16 I/O pins normally referred to as P0, P1, to P15 as shown in figure 2.2. The I/O pins are described as bi-directional, which means each pin can be an input **or** an output.
However, a pin **cannot be both (input and output) at the same time.** The pin's direction has to be configured in the program. The configuration determines how the Micro-Controller should handle a specific pin.
The Basic Stamp Micro-Controller will monitor the status of an input configured pin, and will control the status of an output configured pin.

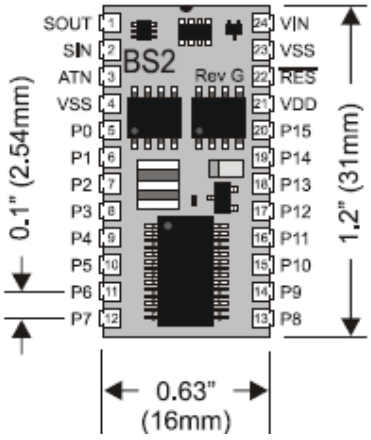


Fig 2.2: Basic Stamp pin 's assignments

I/O related Commands

There are many available commands to determine the behavior of a pin such as INPUT, OUTPUT, REVERSE, HIGH, LOW, and

TOGGLE commands. Detailed information about those commands and their usage is mentioned here after:

OUTPUT command

This command makes a specified pin output**.**
Command Syntax: **OUTPUT** pin
Pin: is the required pin number (0-15).
Example: **OUTPUT** 1
Set pin number 1 to be an output.

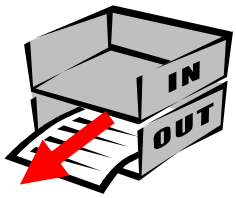


Fig 2.3: OUTPUT command makes a specified pin output

INPUT command

This command makes a specified pin input**.**
Command Syntax: **INPUT** pin
Pin: is the required pin number (0-15).
Example: **INPUT** 6
Set pin number 6 to be an input.

Fig 2.4: INPUT command makes a specified pin input

Reverse command

This command reverses the pin direction; if pin is input, it will be reversed to be output and vise-versa.
Command syntax: **REVERSE** pin
Pin: is the required pin number (0-15).
Example: **REVERSE** 15
Change the direction of pin 15
If input → change to output
If output→ change to input

HIGH command

This command makes a specified pin output, and sets its status to high **(on)**
Command syntax: **HIGH** pin
Pin: is the required pin number (0-15).
Example: **HIGH** 5
Configure pin 5 to be output and set its status **on**.

Fig 2.5: HIGH command makes output pin on

LOW command

This command makes a specified pin output, and set its status to low **(off)**
Command syntax: **LOW** pin
Pin: is the required pin number (0-15).
Example: **LOW** 5
Configure pin 5 to be output and set its status **off.**

Fig 2.6: LOW command makes output pin off

TOGGLE command

This command makes a specified pin output and inverts its status
If status high → change to low
If status low → change to high
Command syntax: **TOGGLE** pin
Pin: is the required pin number (0-15).
Example: **TOGGLE** 13
Configure pin 13 to be output and invert its status.

Outputs Control: LEDs

The Basic Stamp Micro-Controller can be used in electrical circuits to control the supply of power to certain components such as LEDs. This will be much easier than manually changing the wiring or connecting and disconnecting the power supply. Even better, the power control can be done in a smart way. This can be achieved via programming

For example the Basic Stamp can be programmed to do the following:

- Turn LED on and off at different rate
- Turn LED on and off a certain number of times
- Control more than one LED in one electrical circuit

Fig 2.7: Output LEDs in devices are controlled by Micro-Controllers

Electrical Basics Refreshment: Resistors

Do you remember resistors, and what resistors do in a circuit? Resistors, from the name, resist the flow of current. The resistance unit is ohm (Ω). The resistor's schematic diagram is shown in figure 2.8, and it looks like figure 2.9 in reality. The color stripes on the resistor will inform you its resistance value.

Fig 2.8: Resistor schematic representation

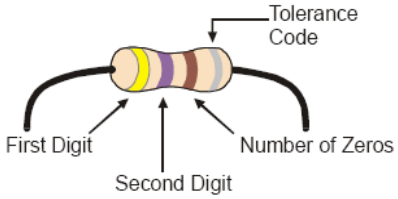


Fig 2.9: Resistor in reality

Each color bar corresponds to one digit of the resistance value; the color code is listed in table 2.1.
In the resistor, the first and second color bars represent the first and second digits. The third color bar represents the number of following zeros.
There may be a fourth color bar that indicates the resistor's tolerance. The tolerance tells you how far off the part's true resistance might be from the labeled resistance. Tolerance is measured in percent, and it could be gold (5%),

Color	1st Digit/ 2nd Digit	3rd Digit (No. of 0s)
Black	0	
Brown	1	0
Red	2	00
Orange	3	000
Yellow	4	0000
Green	5	00000
Blue	6	000000
Violet	7	0000000
Gray	8	00000000
White	9	000000000

Table 2.1: Resistor Color Code

silver (10%) or no stripe (20%).
Can you find out the value of the resistor shown in figure 2.10?
The color of the 1st bar is **yellow**, which corresponds to the 1st digit, the digit is _____
The color of the 2nd bar is **violet**, which corresponds to the second digit, the digit is _____
The color of the 3rd bar is **brown**, which corresponds to the number of zeros to follow the 2nd digit. The number of zeros is _____
Therefore the resistance value calculated is _____Ω.

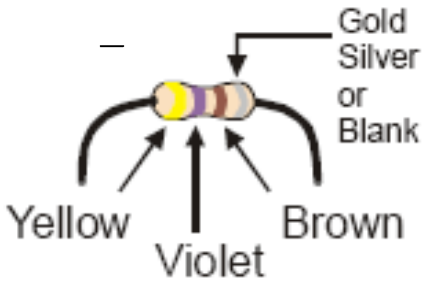


Fig 2.10: Resistor color bars determine its resistance value

LED:

A light emitting diode (LED) emits light when the current passes though it in a certain direction; it has no response when the current passes through the other direction. Therefore it is very important to connect the LED terminals in the right direction. The longer terminal, called "anode", is the (+) terminal. The (-) shorter terminal is called cathode


Fig 2.11: LED in reality

Fig 2.12: LED Schematic diagram

LED Test Circuit Experiment

In this experiment you will build your electric circuit consisting of LED, and 470 Ω a resistor on the BOE solderless breadboard and test it without, then with connection to Basic Stamp Micro-Controller.

How to use breadboard to connect circuits?

As shown in figure 2.13, the solderless breadboard consists of two groups of rows. In each group every row consists of five **interconnected** sockets. The sockets of the same row of the two groups are not connected with each other.

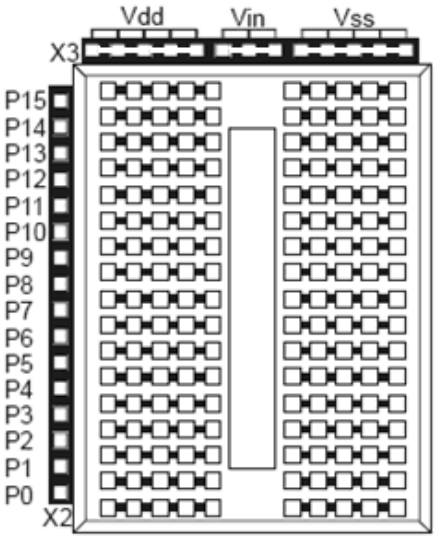


Fig 2.13: Solderless breadboard in BOE

The black sockets on top of the breadboard are used to supply power to the built electric circuit and are referred to as **supply header.**
We will not use Vin in our experiments.

The black sockets at the side are used to connect to the I/O pins of the Basic Stamp Micro-controllers and referred to as **pins header.**

Vdd stands for _____.
Vss stands for _____ .
Vin stands for _____

LED test circuit without Basic Stamp

Set the 3-position switch on the Board of Education to position-0 to turn off the Power of the board of Education.

Fig 2.14: Power switch is off on 0 position

- Connect the cathode terminal of the LED to Vss, and connect the anode terminal into the socket.

- Connect one terminal of the 470 Ω resistor to the anode of he LED, i.e any socket in the same row in the same group. And the other terminal to Vdd. The schematic diagram of the circuit you have connected is as shown in figure 2.15

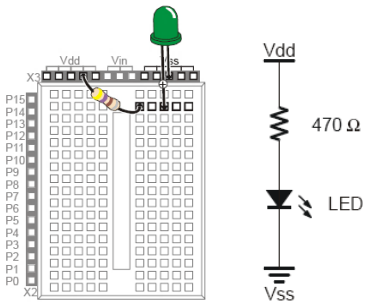


Fig 2.15: LED circuit connection in BOE; circuit schematic diagram.

- Turn on the Power of the board of Education. What can you see?

_____
_____
_____
_____
_____

Can you calculate the current passing through the resistor
Hint: Ohm's low (V/I=R)

Troubleshooting:

If the LED does not glow, investigate the following potential causes:
1- Loose wire; check all wires
2- Direction of LED connection; reverse the direction of the connection
3- Damaged LED; connect another LED

- Turn off the power of the BOE.

- Replace the 470Ω resistor by 2KΩ,
  What is the color code of the 2KΩ resistor?
  1st digit color is _____
  2nd digit color is _____
  Number of digits is _____, the color is _____

- Turn on the power of the BOE. What do you realize? _____
  What is the current passing through the 2KΩ resistor? _____

- Turn off the power of the BOE.

- Replace the 2kΩ resistor by 220Ω,
  What is the color code of the 220Ω resistor?
  1st digit color is _____
  2nd digit color is _____
  Number of digits is _____, the color is _____

- Turn on the power of the BOE. What do you realize? _____
  What is the current passing through the 220Ω resistor? _____

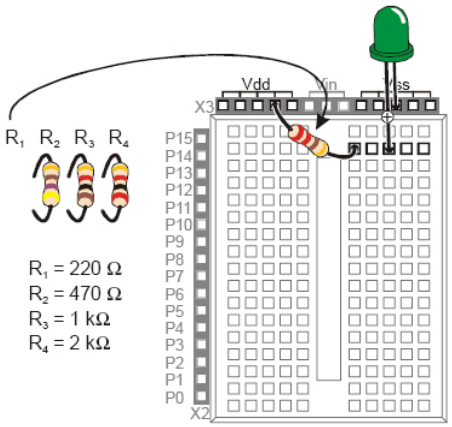


Fig 2.16: Changing resistance value effect on LED

LED test circuit with Basic Stamp

In this part of the experiment you will connect the Basic Stamp Micro-Controller to the LED circuit. You will turn on/off the LED via Basic Stamp.

- Set the 3-position switch on the Board of Education to position-0 to turn off the Power of the board of Education.

Fig 2.17: Power switch is off on 0 position

- Modify the circuit connection as per figure 2.18. Use **470Ω** resistor.

  It is very important to use a resistor to protect the Micro-Controller from any damage that could be caused by over-current.

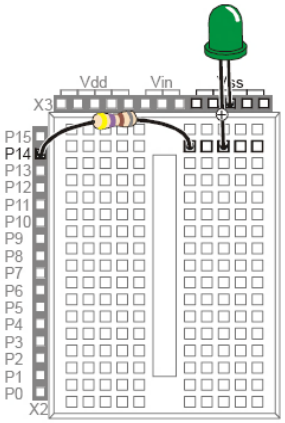


Fig 2.18: LED test circuit

- Connect the COM port of the serial cable directly to your Board of Education; connect the USB end of the cable to your PC; turn on the Power of the board of Education.

- Open Basic Stamp Editor Software and write the program shown in figure 2.19 into the Basic Stamp.

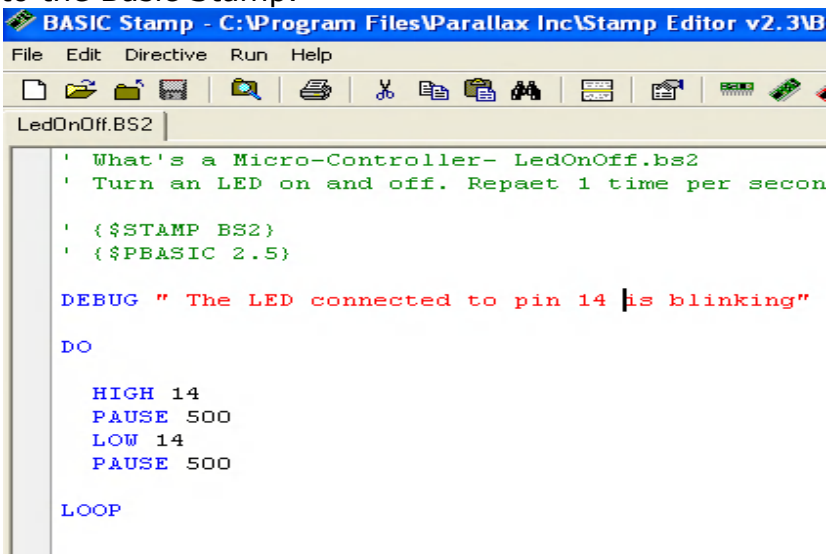


Fig 2.19: LED control program

- Save the Program as LedOnOff.bs2 and download.

1. What can you see?
2. Time the LED on time.
3. Time the LED off time.
4. What would the command **DEBUG** do?
5. What does the command **HIGH** mean?
6. What does the command **LOW** mean?

The command **PAUSE 500** causes the Basic Stamp to pause (freeze) for a **500 ms**. Ms is short for millisecond, which is 1/1000 of 1 second.

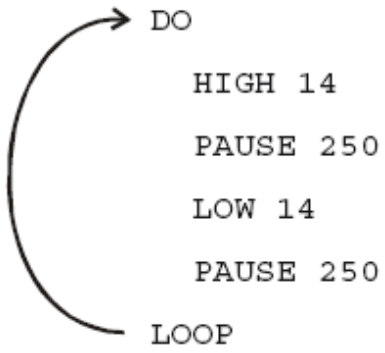


Fig 2.20: DO… LOOP program

- Change the time to pause in the program as shown in figure 2.21. What do you realize?
  The on time is _____s
  The off time is _____s

```
DO

  HIGH 14
  PAUSE 1000
  LOW 14
  PAUSE 2000

LOOP
```

Fig 2.21: Change pause time

- Change the on time pause to 1.5s and the off time pause to 0.25s. Write the correct program code.

DO

_____

_____

_____

_____

LOOP

- Once you are done, do the following:

1- Save your program.
2- Switch off the power of the BOE.
3- Disassemble your circuit, and return the components to their safe.
4- Clean your working area and report any damaged components.

Numbering Systems

In our daily life for any calculation we use a numbering system known as decimal system. Which means counting based on 10:

1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
…

Do you have any idea why 10 in particular? Check figure 2.22 for a hint.
In the decimal system we use the basic digits 0, 1, 2 … 9, that any number will be either one or a combination of those digits such as the number 350.

Fig 2.22 Decimal system base is number 10

350 can be thought of as:
3x100 + 5x10 + 0x1 **or**
$3x\ 10^2 + 5x\ 10^1 + 0x10^0$

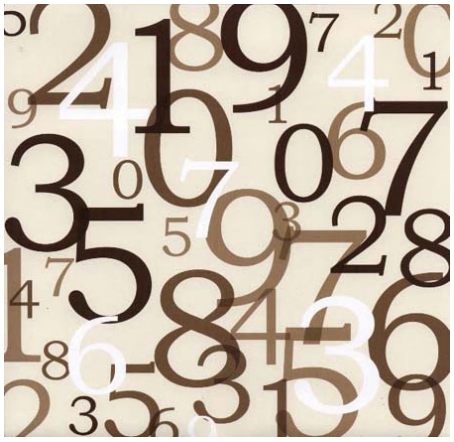


Fig 2.23 Decimal system digits

3,5, and 0 are **digits**;
10 is the **base**;
And the powers 0,1 and 2 determines the positional **weight** of the digit.

Actually any number can be categorized by its digits, base, and weight.

Can you find out the digits, and weights of the number 179?

_____
_____
_____

Note that the largest 3-digit number will be 999. The largest 7-digit number would be 9999999.

Let us assume that we live in an imaginary world where people have only one hand of 5 fingers. 1,2,3,4,10
They will be using a base 5 numbering system. Can you find out which decimal number is equivalent to the base 5 number (41)?

41 (base 5 number):
The digits are: 4, 1
The Base is: 5

→ $4x5^1 + 1x5^0$
→ 20 + 1
→ 21 (base 10)

Can you find out the equivalent decimal number of number 10 (base 5)? _____

_____

What is the largest 4-digit number in the base 5 numbering system? _____

Binary System

In the digital world, input and output devices are controlled by off and on commands, i.e. 0 or 1. Therefore the binary numbering system of the **digits 0, 1 and base 2** is very commonly used in that world.

Fig 2.24: Binary system digits are 0 and 1.

A typical 4-digit binary number would be 1010, which does not mean decimal one thousand and ten. Can you find out the decimal value of this number?
Binary 1010 →
$1x2^3 + 0x2^2 + 1x2^1 + 0x2^0$ →
8 + 0 + 2 + 0 →
Decimal 10

Now you know why there are lot's of 0s and 1s in the movie Matrix.

Programming Commands (PAUSE, LOOP commands)

Variables

Before we proceed with the commands, now is a proper time to introduce Variables.
A **variable** is a symbolic name you choose to represent a value in the memory.

Variable Type	Range of Values
Bit	0-1
Nib	0-15
Byte	0-255
Word	0-65535

Table 2.2: Variable Types

Variable Types

The memory location of the variable itself does not matter as it will be random in RAM. But what really matters is the variable type. Variable types are listed in table 2.2.
The variable type is determined by the

maximum value a variable could be as it determines the size the memory that should be reserved for it.
Here below are some examples:

- Let's say that we have a variable that will represent on and off state of a switch; we will name that variable **"switch",** the variable type will be **bit** because it is either 0 (Off) or 1 (on) only.

- Let's assume you want to write a program to count the number of cars entering a mall in one day. You know that maximum number of cars will never exceed the value 65535, but would be possibly more than 255 cars. So you "declare" a variable named **cars** or possibly **counter**. The variable type in this case would be **word.**

Fig 2.25: The variable "Switch" of type bit can be on (1) or off (0)

Hints:
- **All the terms bit, nib, byte, and word are coming from the binary system.**
- **A bit is a binary digit that could be 0 or 1 only.**
- **A nib is 4-bit number; it consists of 4 digits only.**
- **A byte is 8-bit number; it consists of 8 digits only**
- **A word is 16-bit number; it consists of 16 digits only.**

Now the one million Dirhams question is what is the significance of the numbers 15 in a nib, 255 in a byte, and 65535 in the word when defining the data type of a variable?
Well, Here are some hints.
Could you guess the answer yet? Well, then have a look at some more hints.

More Hints:
- **The maximum value of a binary 4-bit number is 1111. the decimal value of this number is:**
  →$1x2^3 + 1x2^2 + 1x2^1 + 1x2^0$
  → 8 + 4 + 2 + 1
  → 15 (Nib)

- **The maximum value of a binary 8-bit number is 11111111. Calculate the decimal value of this binary number.**
  _____.

- **The maximum value of a binary 16-bit number is 1111111111111111. Guess the decimal value of this binary number.**
  _____.

Variable Declaration

Before using the variable in a program you have to declare it first. Declaration allows the memory to allocate the right memory size for the variable

Variable declaration syntax:
Symbolic name VAR Type
Symbolic name is the name you choose for the variable.
VAR is short for variable to indicate a declaration in process.
Type is variable type, which could be bit, nib, byte, or word.

Examples:
Counter VAR word
Declaring a variable named Counter of type word.
Switch VAR bit
Declaring a variable named switch of type bit.

Symbolic names selection rules

There are some rules to follow when selecting a variable symbolic name:

- The symbolic name must start with a letter; it may contain letters, numbers, or underscore.
- The name cannot have a space.
- The name cannot exceed 32 characters long.
- The name cannot be a word that is already used by Basic Stamp (DEBUG, PAUSE, HIGH, LOW, DO, and LOOP). These words are called reserved words

Using a variable in a program

There are many ways to use a variable in a program; one possible way is to display the variable value in DO… LOOP

program as shown in figure 2.26.
Some other ways shall be covered in the coming sections.

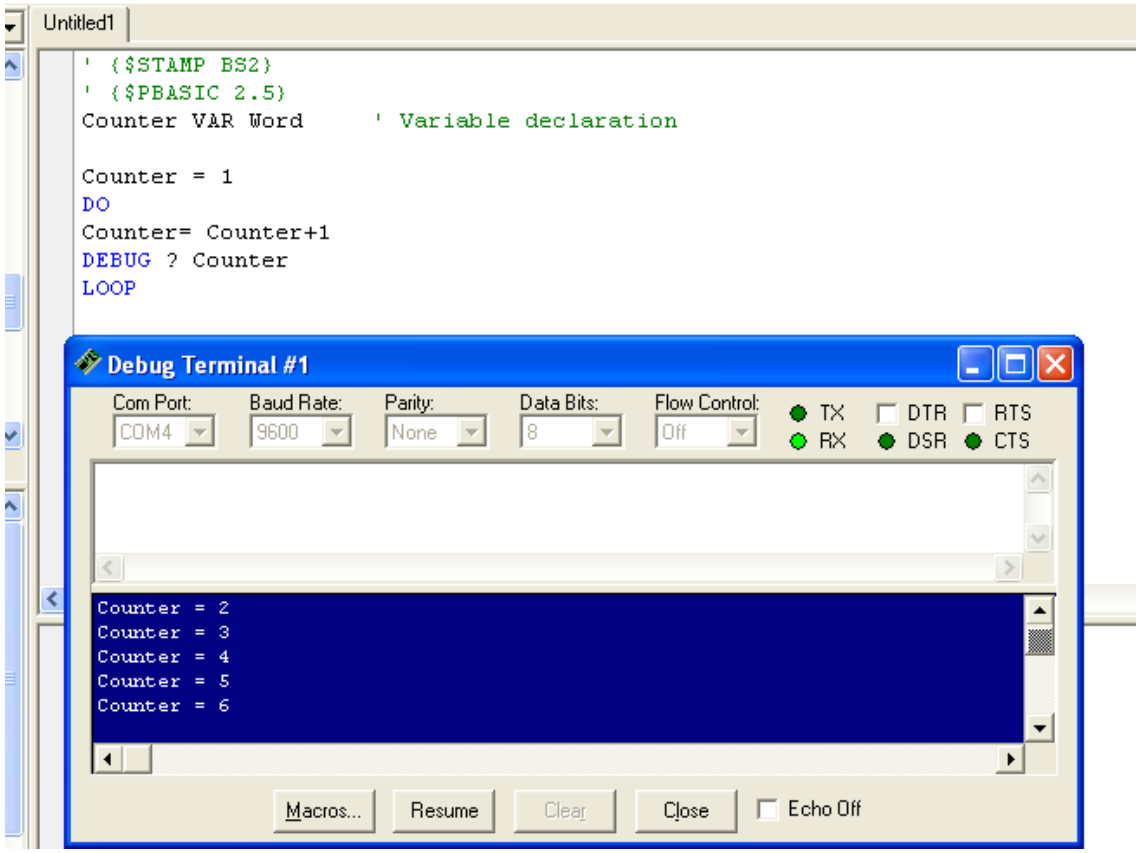


Fig 2.26: Example of using a variable in DO … LOOP program

PAUSE Command

As you know by now this command delays the execution of the next command for the specified duration of time.
Command syntax: **PAUSE** duration
Duration: is duration of the pause in millisecond. The maximum duration value is 65535ms.
Find out how many seconds that is: _____
Find out how many minutes that is: _____
Example: **PAUSE** 1000
Wait for 1000ms before execute the next command.
1000ms= 1000/1000s=1s.

Fig 2.27: Pause commands stops the Program for a predefined time

DO … LOOP Command

It is a loop command that lets the program execute the commands sandwiched within indefinitely. Once the Program recognizes the term LOOP, it will jump back to term DO.
Command syntax:

```
DO
…. ' Other commands
…
…
LOOP
```

Example:

```
DO
DEBUG "Test", CR
PAUSE 1000
LOOP
```

Once the Program recognizes the term DO, it marks the LOOP starting point. The DEBUG command will display "Test" on the PC; CR means insert new line on the screen. The PAUSE command will force the program to wait before it executes the next command for 1 second. After the time elapses it shall

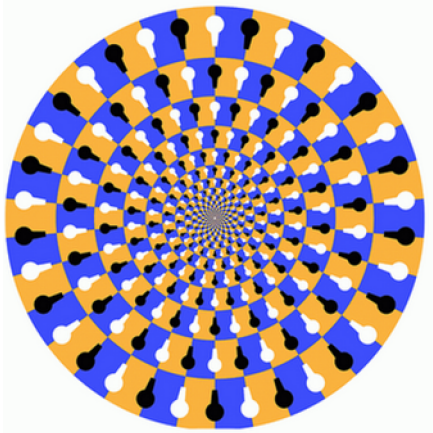


Fig 2.28: Do… LOOP commands cause the program to loop over and over and over

execute the command LOOP. LOOP command will cause the program to jump back to the command DO. The mentioned steps will repeat again

DO WHILE … LOOP

While DO… LOOP is unconditional loop that will keep on looping indefinably, DO… While is a conditional looping command that the program will execute as long as some conditions are **true**
Command syntax:
DO WHILE (condition)
…
…
LOOP

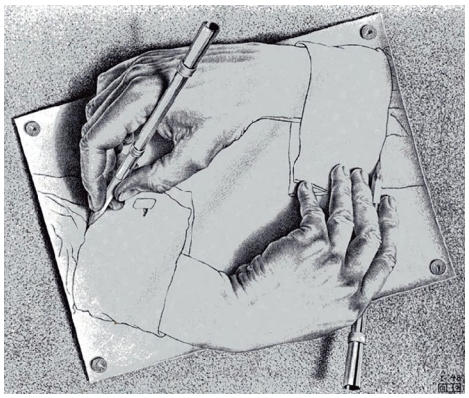


Fig 2.29: DO … WHILE is another LOOP command

Example

```
Counter VAR word
Counter=1
DO WHILE (Counter <= 255)
Counter=Counter+1
DEBUG ? Counter
LOOP

DEBUG  "LOOP Terminated"
```

The word type variable **Counter** was declared and set to value 1.
At DO WHILE step, the program will evaluate counter and it will check if it is less than or equals to 255. If this condition is true it will continue to next line. The counter will be incremented by one. The DEBUG command will display the counter value on the PC screen.
The same steps will loop over and over until the counter value becomes 256.
Once it be comes 256 it will skip the DO-LOOP and move to the next command.
"Loop terminated" message will appear on the screen.

DO.. LOOP UNTIL

DO … LOOP UNTIL is a conditional looping command that the program will execute as long as some conditions is **not true yet.**

Command Syntax**:**
DO
…
…
LOOP UNTIL (condition)

Example
Counter VAR word
Counter=1
DO
Counter=Counter+1
DEBUG ? Counter
LOOP UNTIL (Counter >= 255)
DEBUG "LOOP Terminated"

The word type variable **Counter** was declared and set to value 1.
At the step of DO, the program will recognize the start point of the loop. The counter will be incremented by one. The DEBUG command will display the counter value on the PC screen. At LOOP UNTIL Step, it will check if the counter is greater than or equals to 255 conditions. If not it will jump back to DO step.
The same steps will loop over and over until the counter value becomes 255.
Once it be comes 255, condition becomes true, it will move to the next command
"Loop terminated" message will appear on the screen.

FOR … NEXT

Fig 2.30: FOR … NEXT controls the number of loops

FOR … NEXT is another kind of looping commands that you can use in the program to specify and monitor the number of loops by what can be considered an incrementing or decrementing counter.
Command syntax:
FOR counter= start value TO end value STEP
…..
…..
Next

FOR marks the beginning of the loop

Counter is the variable you create to control the number of loops.

Start value is a number (0- 65535) that specifies the initial value of the counter

End value is a number (0- 65535) that specifies the end value of the counter.

Step is a number (0-65535) that determines the step the counter value should increase or decrease every time. It is optional and in case not mentioned the step value will be by default 1.

Examples
Cars VAR word
For Cars=1 **TO** 10
DEBUG ? Cars, CR
Next

The variable named cars was declared.
For marks the start of loop; the variable Cars initial value will be 1 and will increment every loop by 1 to final value 10.
DEBUG command will display the current

Cars value.
At NEXT step if the counter value is less than 10, the program will jump back to FOR step. If the Cars value equals to 10 it will terminate the loop and proceed to next commands if any.

```
Cars VAR word
For Cars=0 TO 20 STEP 5
DEBUG ? Cars, CR
Next
```

The variable named Cars was declared.
For marks the start of loop; the variable Cars initial value will be 0 and will increment every loop by 5 to final value 20.
DEBUG command will display the current Cars value.
At NEXT step if the counter value is less than 20, the program will jump back to FOR step. If the Cars value equals to or greater than 20 it will terminate the loop and proceed to next commands if any.

LED Test Experiment # 2

- Set the 3-position switch on the Board of Education to position-0 to turn off the Power.

Fig 2.31: Power switch is off on 0 position

- Modify the circuit connection as per figure 2.29. Use **470Ω** resistor.

  It is very important to use a resistor to protect the Micro-Controller from any damage that could be caused by over-current.

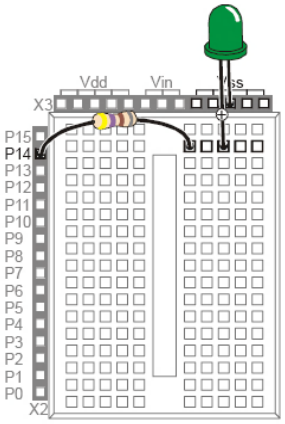


Fig 2.32: LED test circuit

- Connect the COM port of the serial cable directly to your Board of Education; connect the USB end of the cable to your PC; turn on the Power of the board of Education.
- Open Basic Stamp Editor Software.

- Write a program to make the LED blinks On/Off for 10 times. The On time equals 1s; the Off Time equals .5s using FOR … NEXT command.

- Download the program, and test the results.

- Write a program that does the same function described above by using DO WHILE command. Download the Program and test the results.

- Disconnect the power from your BOE. Construct the second LED circuit as shown in figure 2.33. The resistor value is also 470 Ω.

- Reconnect power to the BOE and check the second LED status.

- Disconnect the power from your BOE again, and connect the 2nd resistor to P15.

- Write the following program into the Micro-Controller's software.

```
DEBUG "Program Running"
DO
HIGH 14
HIGH 15
PAUSE 500
LOW 14
LOW 15
PAUSE 500
LOOP
```

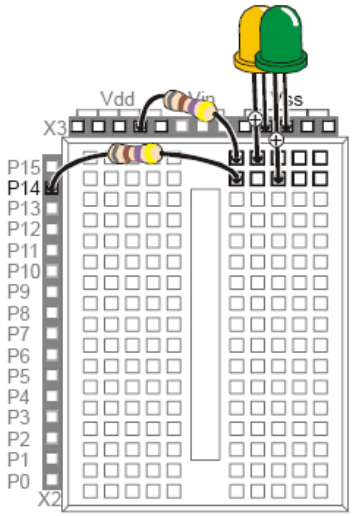


Fig 2.33: 2x LEDs test circuit

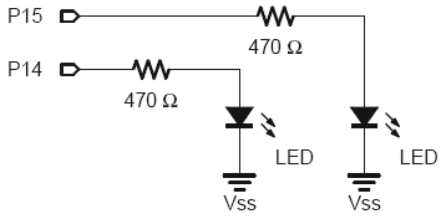


Fig 2.34: the circuit schematic diagram

- Write your observations

_____
_____
_____
_____
_____
_____
_____
_____

- Can you write a program that when LED is on, the other LED is OFF and vise versa?

_____
_____
_____
_____
_____
_____
_____
_____

Bi-Color LED Test Experiment

The Bi-Color LED is an LED that can emit two colors. The color alternates based on the voltage supply direction. If you connect the green-red Bi-Color LED shown in figure 2.35 in a certain direction, it will glow green, and if you reverse the connection it will glow red.

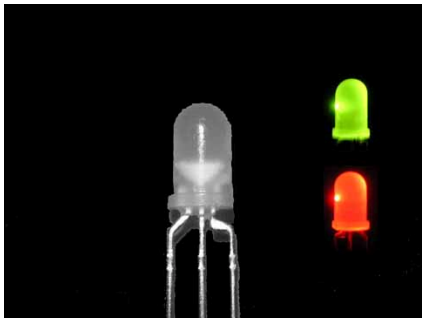


Fig 2.35: Green Red Bi-Color LED

The technology driving the LED behavior is that there are two reversed direction LEDs and different in color connected in parallel. Remember that LED acts as one way current valve. So if one diode allows the current to pass the other blocks it. So the two LEDs are never on at the same time. The LED schematic diagram is shown in figure 2.36

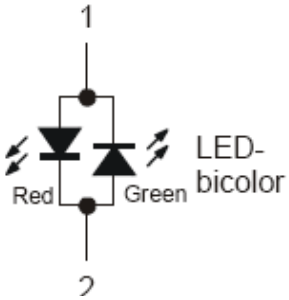


Fig 2.36: Bi-Color LED Schematic diagram

- Disconnect the power from BOE.
- Build the circuit shown in figure 2.37 using a 470 Ω resistor and the Bi-Color LED.
- Reconnect the power. What is the LED color? _____
- Disconnect the power again, and reverse the direction on LED connection (flip the LED)
- Reconnect the power. What is the LED color? _____

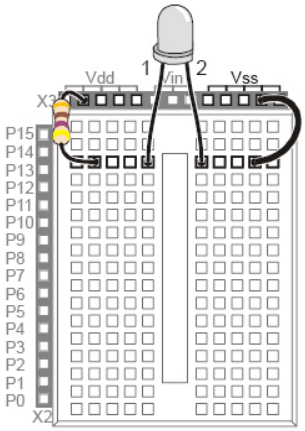


Fig 2.37: Bi-Color test circuit

In order to control the Bi-Color LED by Basic Stamp Micro-Controller you need to connect both ends of the circuit to I/O pins.

- Disconnect he power of BOE
- Build the circuit shown in figure 2.38
- Write the following program, download, and test the results.

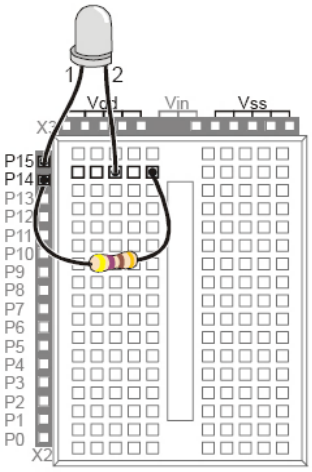


Fig 2.38: Bi-Color connection to Basic Stamp circuit

```
DEBUG "Program Running!"

DO

HIGH 15
LOW 14
PAUSE 500    'LED  is _____

LOW 15
HIGH 14
PAUSE 500    'LED is _____

LOW 15
LOW 14
PAUSE 500    'LED is _____
LOOP
```

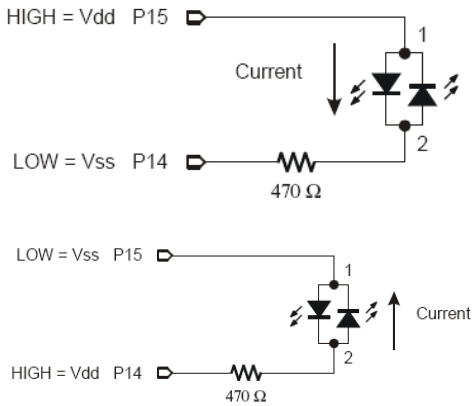


Fig 2.39: Circuit Schematic diagram.

- Write the following Program, download it, test and analyze

```
Counter VAR BYTE

FOR counter= 1 TO 50
HIGH 15
LOW 14
PAUSE  Counter

LOW 15
HIGH 14
PAUSE Counter

NEXT
```

The variable named _____ Was created, and it is of type _____.

The maximum value of the that variable could be _____

the number of loops, the LED will is _____

What happened when the pause time was set to the variable counter?

_____
_____
_____

- Modify the program to the following:

What is the difference?

_____
_____
_____
_____
_____
_____
_____

```
Counter VAR BYTE

DO

FOR counter= 1 TO 50
HIGH 15
LOW 14
PAUSE  Counter

LOW 15
HIGH 14
PAUSE Counter

NEXT

LOOP
```

- Once your done

1- Save your program
2- Switch off the power of the BOE.
3- Disassemble your circuit, and return the components to their safe.
4- Clean your working area and report any damaged components

Inputs Monitoring: Pushbuttons

How many devices with pushbuttons do you use on a daily basis? Here are a few examples that might appear in your list: computer, mouse, calculator, microwave oven, handheld remote, handheld games, and VCR. In each device, there is a microcontroller scanning the pushbuttons and waiting for the circuit to change. When the pushbutton status changes, the Micro-Controller detects the change and takes action.

Fig 2.40: Keyboard with lots of pushbutton keys

The drawing and the schematic diagram of the normally open pushbutton that will be used in the coming experiments is illustrated in figure 2.37. We will use PB as a short for pushbutton for ease of use. As you can see, in the drawing the pushbutton has four pins numbered 1, 2, 3, and 4. Pins 1 and 4 are internally connected, and pins 2 and 3 are internally connected which explains why the schematic diagram has two pins only. The reason the pushbutton doesn't just have two pins is because it needs stability. If the pushbutton only had two pins, those pins would eventually bend and break from all the pressure that the pushbutton receives when people press it.

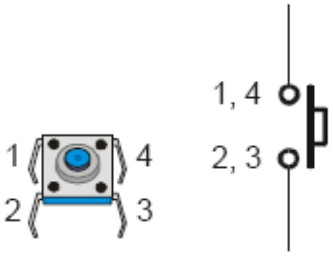


Fig 2.41: Pushbutton in reality (left) and schematic diagrams (right).

When the PB is not pressed, there is an open circuit state, which means that current cannot flow from pins (1,4) to (2,3) due to the gap between them. When the button is pressed, the gap is bridged by a conductive metal that allows the current to flow to form a closed circuit state

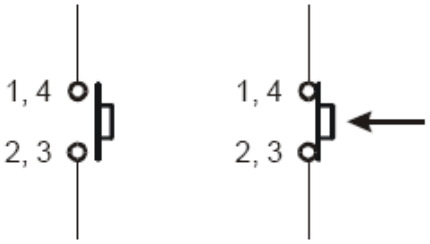


Fig 2.42: Pushbuttons Open status (left) and closed status (right).

Pushbutton Test Circuit Experiment

In this experiment you will build an electrical circuit consisting of LED, 470Ω resistor and a PB as will as a jumper wire. You will test the functionality of the PB with and without connection to Basic Stamp Micro-Controller.

Pushbutton Test circuit without Basic Stamp

- Set the 3-position switch on the Board of Education to position-0 to turn it off.

Fig 2.43: Power switch is off on 0 position

- Build the circuit shown in figure 2.44. Pay attention to the LED direction and the PB pins connection.

  Can you draw the circuit schematic diagram in the box in below.

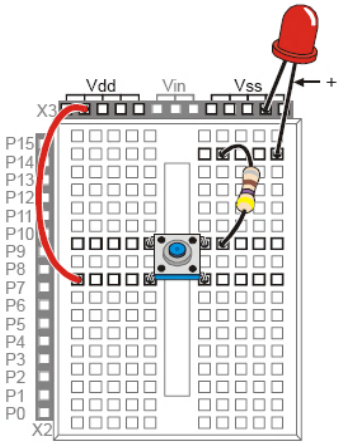


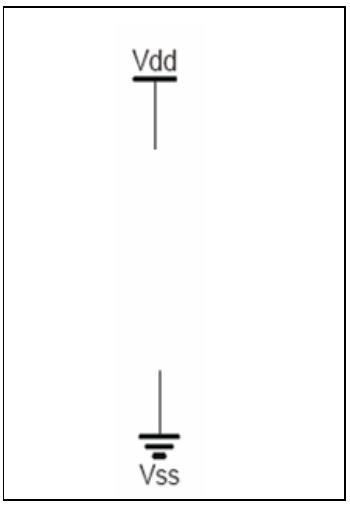
Fig 2.44: circuit connection

- Reconnect the power, what do you realize?

  When the PB is not pressed the LED is _____.

  When the PB is pressed the LED is _____.

  Press and hold the PB, the LED is _____.



If the Power LED on the Board of Education flickers, goes dim, or goes out completely, it means you have a short circuit. If this happens, disconnect power immediately and find and correct the mistake in your circuit.

- Disconnect the power.
  Modify the circuit as shown in figure 2.45.
  Can you draw the circuit schematic diagram in the box below.

- Reconnect the power, what do you realize?

  When the PB is not pressed the LED is _____.

  When the PB is pressed the LED is _____.

  Press and hold the PB, the LED is _____.

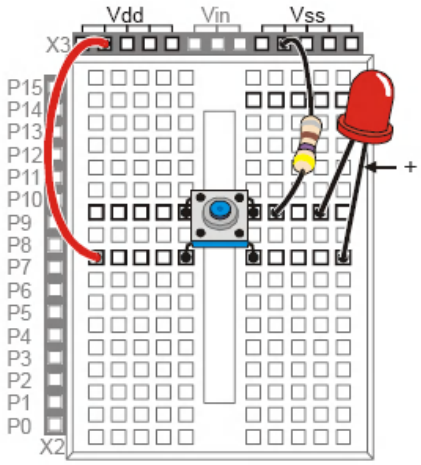
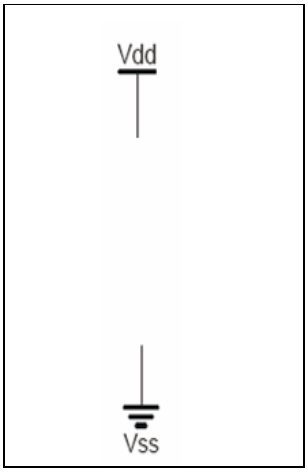


Fig 2.45: Modified circuit



The reason the LED turns off when the pushbutton is pressed is because current always follows the path of least resistance. When the pushbutton is pressed, terminals (1,4) and (2,3) have almost no resistance between them, so all the current passes through the PB (short circuit) instead of the LED.

Pushbutton test circuit with Basic Stamp

In this part of the experiment you will connect the Basic Stamp Micro-Controller to the PB electrical circuit. The Micro-Controller will monitor the status of the PB and display its value on the PC screen. To build the circuit, other than the PB, you will need 220Ω, and 10kΩ resistors (one each), and few jumper wires.

- Disconnect the power.
- Build the circuit as shown in figure 2.46. The schematic diagram shown in figure 2.47 will help you with connection as well.

When the pushbutton is pressed, the BASIC Stamp senses that Vdd is connected to P3. Inside the BASIC Stamp, this causes it to place the number 1 in a part of its memory that stores information about its I/O pins. When the pushbutton is not pressed, the BASIC Stamp cannot sense Vdd, but it can sense Vss through the 10 kΩ and 220 Ω resistors. This causes it to store the number 0 in that same memory location that stored a 1 when the pushbutton was pressed. This memory location can be accessed by the IN command

Command syntax: INpin
Example: IN3; which means read the status of pin 3.

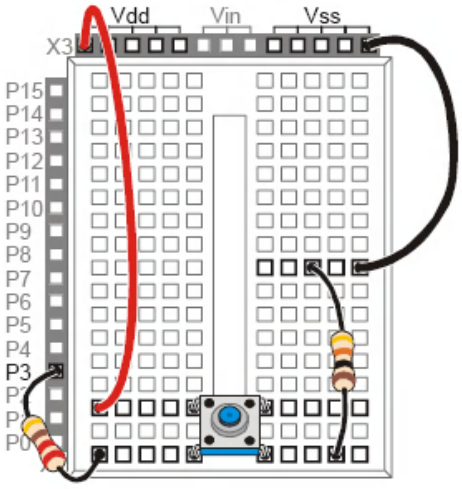


Fig 2:46 Pushbutton circuit connection

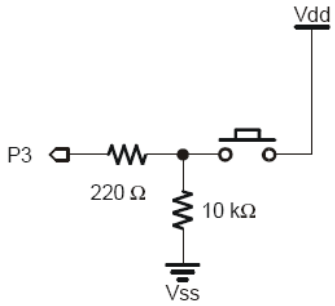


Fig 2:47 Circuit schematic diagram

- Reconnect the power of the BOE.
- Write the following program, download and test.

```
DO
DEBUG IN3
PAUSE 250
LOOP
```

When the PB is pressed, IN3 value displayed is _____.

When the PB is not pressed, IN3 value displayed is _____.

What does the program do?

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

- Disconnect the power of the BOE.
- Modify the circuit as shown in figures 2.48 and 2.49,
- Download the previous program and test it

When the PB is pressed, IN3 value displayed is _____.

When the PB is not pressed, IN3 value displayed is _____.

Do you know why did the circuit behave in the manner it did in the experiment? The answer lies in the basics of electrical circuits.
Looking at the schematic diagram in figure 2.49, when the PB is not pressed (open circuit condition), the full amount of the current will flow from Vdd to P3, i.e. the value read is 1. When the PB is pressed and because of its almost zero resistance, it acts as a short circuit and the full amount of current will pass through. P3 will read zero value in this case.

- Once your done

1- Save your program
2- Switch off the power of the BOE.
3- Disassemble your circuit, and return the components to their safe.
4- Clean your working area and report any damaged components

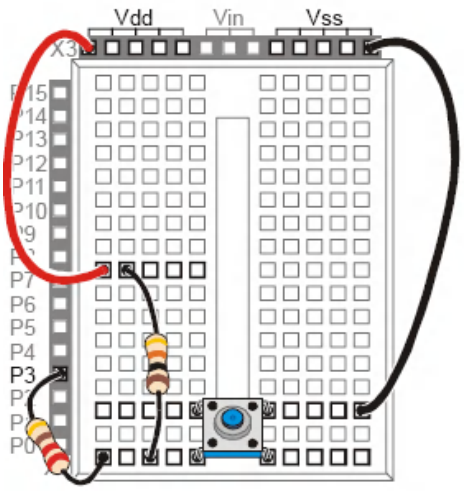


Fig 2.48: Modified pushbutton circuit connection

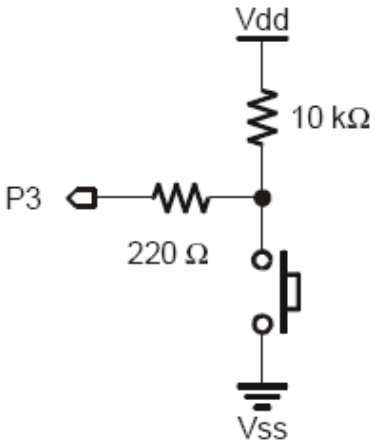


Fig 2.49: Schematic circuit diagram

Programming Commands (IF-THEN)

IF THEN is a command used to evaluate a certain "condition". The evaluation outcome is true or false. If the evaluation outcome is true, a certain action will be taken, if the evaluation outcome is false another action will be taken.
The IF Then statement consists of the 3 parts:

1- Condition: it is a statement to be evaluated (true or false). Here some examples of conditions:
(IN3=0), (cars>255), (cars<>0), or (cars<=255).

2- True arguments: commands to be executed if the condition evaluation turns a "true" outcome.

3- False arguments: commands to be executed if the condition evaluation turns a "false" outcome.

A real live example would be:
If it is night time, then switch on the light; if it is day time, then switch off the light. Another example of IF Then, yet more complicated is the following:
IF you get "F" OR "D" grade in this course, you will loose your car for the entire summer with a curfew stating at 7:30 pm; enjoy being grounded at home
IF you get a "C" grade, you will be only allowed to drive 1995 Toyota Tercel, and pay for the petrol from your allowance.
IF you get a "B" grade, you will get a brand new FJ Cruiser; sorry you still pay for the petrol.
IF you get "A" grade, you will get a new a brand new Ford Mustang, with paid petrol
IF you score the full mark, you will get a custom made Ferrari with paid petrol. I hope the price is worth the efforts. Now quit daydreaming and back to reality.

Fig 2.50: 1995 Toyota Tercel

Fig 2.51: Ford Mustang

Fig 2.52 Ferrari concept

There are ways to express IF THEN statement.

The standard way command syntax:
IF (condition) **THEN**
…
True arguments
…
ELSE
…
False arguments
…
ENDIF

Example
Dark VAR bit
Dark = IN0 ' input 0 provides dark status
IF (Dark=1) **THEN** 'if dark
HIGH 1 'true argument switch on light Output 1
ELSE ' if not dark
LOW 1 ' switch off light Output1
ENDIF
A variable name Dark of type bit was declared and was set equal to input pin 0. If the dark was found to be 1, then switch on the light (high output pin 1). Else, if it was found not dark, switch off the light (low output pin 1).

Another form of IF THEN statement is as follows:
Command syntax:
IF (condition1) **THEN**
…
True arguments for condition 1
…
ELSEIF (condition2) **THEN**
…
True arguments for condition 2
…
ELSE
False arguments for conditions 1&2
ENDIF

Fig 2.53: If day time (top), then off the lights, and if dark (bottom), then on the light.

Example

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

Grade VAR Byte

FOR Grade=50  TO 100 STEP 5
PAUSE 1000
IF(Grade<70) THEN
DEBUG DEC5 Grade,"   Grounded, No car, Extended curfew" ,CR
ELSEIF (Grade<80) THEN
DEBUG DEC5 Grade,"   Enjoy the 1995 TRCEL, pay for petrol", CR
ELSEIF (Grade <90) THEN
DEBUG  DEC5 Grade,"   Congrats; NEW FJ Cruiser. Pay for petrol"  ,CR
ELSEIF(Grade <=99) THEN
DEBUG DEC5 Grade,"   Congrats; NEW MUSTANG & Free Petrol" ,CR
ELSE
DEBUG DEC5 Grade,"  Congrats; GRAND FERRARI" ,CR
ENDIF

NEXT
```

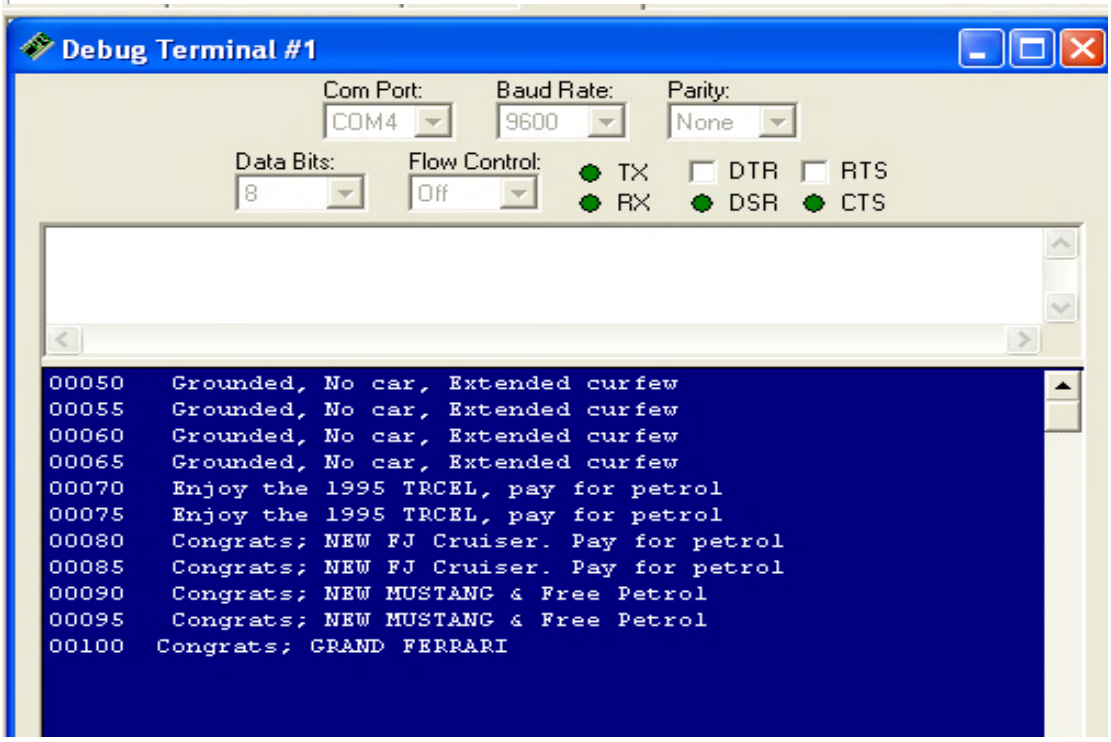


Fig 2.54: IF … NEXT Program and Debug Terminal window

"DEBUG DEC5 Grade" means display the decimal value of the variable Grade; 5 indicates the number of digits.

The Maximum number of IF THEN statements in one program in 16.

The maximum number of ELSEIFs in one IF THEN Statement is 16.

The maximum number of ELSEs in one IF THEN Statement is 1only.

1 LED Logic controlled Circuit Experiment

In this experiment you will build an electrical circuit consisting of:

1x LED, 1x 470Ω, 1x 220Ω, 1x 10kΩ resistors, 1x PB and 2x jumper wires.

In this experiment, you will program the Micro-Controller to flash the LED based on the PB status.

- Set the 3-position switch on the Board of Education to position-0 to turn off the Power.

Fig 2.55: Power switch is off on 0 position

- Build the circuit shown in figures 2.56, 2.57 and 2.58 Pay attention to the LED direction and the PB pins connection
- Reconnect the power of the BOE.
- Write the following program, download, and test.

```
DO

DEBUG ? IN3, CR
IF (IN3=1) THEN
HIGH 14
PAUSE 50
LOW 14
PAUSE 50
ELSE
PAUSE 100
ENDIF
```

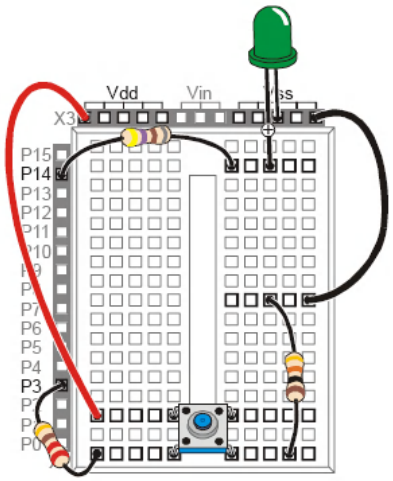


Fig 2.56: PB and LED test circuit

LOOP

- What do you think will happen?

_____

_____

_____

_____

_____

After testing, if you got different results that your expectations write what was different and explain why:

_____

_____

_____

_____

- Modify the program so that the LED flashes twice as fast when you press the PB. Modify the program so that the LED flashes as half as fast

- Once you are done, power off the BOE and get ready for the next experiment.

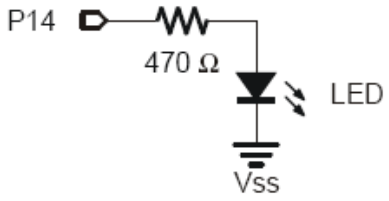


Fig 2.57: LED connection part schematic diagram

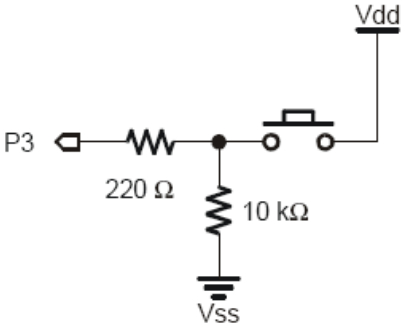


Fig 2.58: PB connection part schematic diagram

2 LEDs Logic controlled Circuit Experiment

In this experiment you will build an electrical circuit consisting of:

2x LED, 2x 470Ω, 2x 220Ω, 2x 10kΩ resistors, 2x PB and many jumper wires.

In this experiment, you will program the Micro-Controller to flash the LED based on the PB status.

- Build the circuit shown in figures 2.59, 2.60 and 2.61 Pay attention to the LED direction and the PB pins connection
- Reconnect the power of the BOE. Write the following program, download, and test.

```
DO
DEBUG HOME
DEBUG ? IN3, CR
DEBUG ? IN4, CR

IF (IN3=1) THEN
HIGH 14
PAUSE 50

ELSEIF (IN4=1) THEN
HIGH 15
PAUSE 50

ELSE
PAUSE 50

ENDIF

LOW 14
LOW 15

LOOP
```

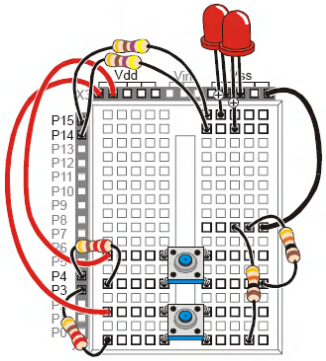


Fig 2.59: 2 LEDs and 2 PBs test circuit

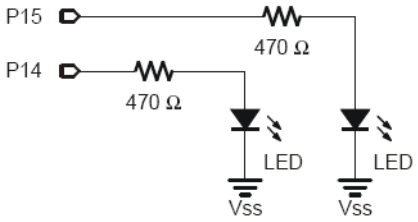


Fig 2.60: LEDs connection part schematic diagram

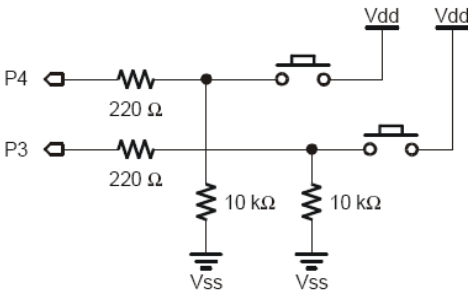


Fig 2.61: PBs connection part schematic diagram

- What happens when the first PB is pressed?

Does it match your expectation? _____

_____
_____
_____
_____
_____

- What happens when the second PB is pressed?

Does it match your expectation? _____

_____
_____
_____
_____

What happens when both PBs are pressed?

Does it match your expectation? _____

This is a bug, can you fix it?

Hint: to combine more than one condition, you can use **"AND"** or **"OR"** connectors. AND connector is used to check if more than one condition is true. All conditions with AND has to be true for the IF statement to be true. OR connector can be used to check if at least one condition is true. The IF command with AND will be as follows:

IF (…) AND (…) THEN

- Modify the program, download, and test.
- Once your are done:

1- Save your program
2- Switch off the power of the BOE.
3- Disassemble your circuit, and return the components to their safe.
4- Clean your working area and report any damaged components

Write the modified program