

# 5: BASIC Stamp Command Reference – INPUT

## INPUT

BS1	BS2	BS2e	BS2sx	BS2p	BS2pe	BS2px
-----	-----	------	-------	------	-------	-------



### INPUT Pin



NOTE: Expressions are not allowed as arguments on the BS1. The range of the PIN argument on the BS1 is 0 – 7.

### Function

Make the specified pin an input.

- **Pin** is a variable/constant/expression (0 – 15) that specifies which I/O pin to set to input mode.

### Quick Facts

Table 5.41: INPUT Quick Facts.

	BS1	All BS2 Models
Input Pin Variables	PINS; PIN0 through PIN7	INS; IN0 through IN15
Related Commands	OUTPUT and REVERSE	

### Explanation

There are several ways to make a pin an input. When a program begins, all of the BASIC Stamp's pins are inputs. Commands that rely on input pins, like PULSIN and SERIN, automatically change the specified pin to input. Writing 0s to particular bits of the variable DIRS makes the corresponding pins inputs. And then there's the INPUT command.

When a pin is an input, your program can check its state by reading the corresponding INS variable (PINS on the BS1). For example:



INPUT 4

Hold:

```
IF IN4 = 0 THEN Hold ' stay here until P4 = 1
```

The code above will read the state of P4 as set by external circuitry. If nothing is connected to P4, it will alternate between states (1 or 0) apparently at random.

What happens if your program writes to the OUTS bit (PINS bit on the BS1) of a pin that is set up as an input? The value is stored in OUTS (PINS on the BS1), but has no effect on the outside world. If the pin is changed to output, the last value written to the corresponding OUTS bit (or PINS bit

## INPUT – BASIC Stamp Command Reference

---

on the BS1) will appear on the pin. The demo program shows how this works.

### Demo Program (INPUT.bs1)



```
' INPUT.bs1
' This program demonstrates how the input/output direction of a pin is
' determined by the corresponding bit of DIRS. It also shows that the
' state of the pin itself (as reflected by the corresponding bit of PINS)
' is determined by the outside world when the pin is an input, and by the
' corresponding bit of OUTS when it's an output. To set up the demo,
' connect a 10k resistor from +5V to P7 on the BASIC Stamp. The resistor
' to +5V puts a high (1) on the pin when it's an input. The BASIC Stamp
' can override this state by writing a low (0) to bit 7 of OUTS and
' changing the pin to output.

' {$STAMP BS1}
' {$PBASIC 1.0}

Main:
  INPUT 7                      ' Make P7 an input
  DEBUG "State of P7: ", #PIN7, CR

  PIN7 = 0                     ' Write 0 to output latch
  DEBUG "After 0 written to OUT7: "
  DEBUG #PIN7, CR

  OUTPUT 7                     ' Make P7 an output
  DEBUG "After P7 changed to output: "
  DEBUG #PIN7
```

### Demo Program (INPUT.bs2)



```
' INPUT.bs2
' This program demonstrates how the input/output direction of a pin is
' determined by the corresponding bit of DIRS. It also shows that the
' state of the pin itself (as reflected by the corresponding bit of INS)
' is determined by the outside world when the pin is an input, and by the
' corresponding bit of OUTS when it's an output. To set up the demo,
' connect a 10k resistor from +5V to P7 on the BASIC Stamp. The resistor
' to +5V puts a high (1) on the pin when it's an input. The BASIC Stamp
' can override this state by writing a low (0) to bit 7 of OUTS and
' changing the pin to output.

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

Main:
  INPUT 7                      ' Make P7 an input
  DEBUG "State of P7: ",
```

NOTE: This example program can be used with all BS2 models by changing the \$STAMP directive accordingly.

## ***5: BASIC Stamp Command Reference – INPUT***

---

```
BIN1 IN7, CR

OUT7 = 0                                ' Write 0 to output latch
DEBUG "After 0 written to OUT7: ",
    BIN1 IN7, CR

OUTPUT 7                                ' Make P7 an output
DEBUG "After P7 changed to output: ",
    BIN1 IN7
```