PULSIN

BS1	BS2	BS2e	BS2sx	BS2p	BS2pe	BS2px

1 All 2

PULSIN ***Pin, State, Variable***

Function

Measure the width of a pulse on *Pin* described by *State* and store the result in *Variable*.

1

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

- ***Pin*** is a variable/constant/expression (0 – 15) that specifies the I/O pin to use. This pin will be set to input mode.
- ***State*** is a variable/constant/expression (0 – 1) that specifies whether the pulse to be measured is low (0) or high (1). A low pulse begins with a 1-to-0 transition and a high pulse begins with a 0-to-1 transition.
- ***Variable*** **is** a variable (usually a word) in which the measured pulse duration will be stored. The unit of time for *Variable* is described in Table 5.83.

Quick Facts

Table 5.83: PULSIN Quick Facts.

	BS1	BS2	BS2e	BS2sx	BS2p	BS2pe	BS2px
Units in *Variable*	10 µs	2 µs	2 µs	0.8 µs	0.8 µs	2 µs	0.81 µs
Maximum Pulse Width	655.35 ms	131.07 ms	131.07 ms	52.428 ms	52.428 ms	123.6 ms	53.08 ms
Related Commands	PULSOUT and COUNT						

Explanation

PULSIN is like a fast stopwatch that is triggered by a change in state (0 or 1) on the specified pin. The entire width of the specified pulse (high or low) is measured, in units shown in Table 5.83, and stored in *Variable*.

Many analog properties (voltage, resistance, capacitance, frequency, duty cycle) can be measured in terms of pulse durations. This makes PULSIN a valuable form of analog-to-digital conversion.

SPECIFICS OF PULSIN'S OPERATION.

PULSIN will wait, for the desired pulse, for up to the maximum pulse width it can measure, shown in Table 5.83. If it sees the desired pulse, it measures the time until the end of the pulse and stores the result in *Variable*. If it never sees the start of the pulse, or the pulse is too long

(greater than the Maximum Pulse Width shown in Table 5.83) PULSIN "times out" and store 0 in *Variable.* This operation keeps your program from locking-up should the desired pulse never occur.

HOW THE RESULT IS REPORTED.

Regardless of the size of *Variable*, PULSIN internally uses a 16-bit timer. Unless the pulse widths are known to be short enough to fit in an 8-bit result, it is recommended to use a word-sized variable. Not doing so may result in strange and misleading results as the BASIC Stamp will only store the lower 8-bits into a byte variable.

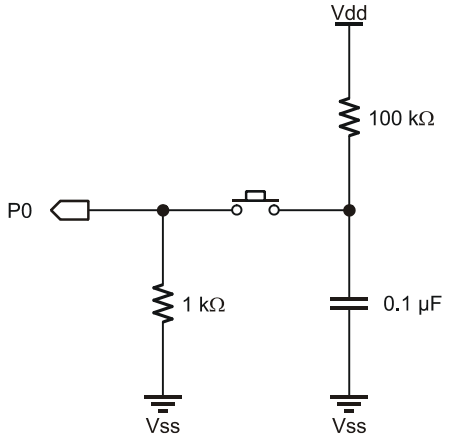

Figure 5.28: R/C Pulse Generator.

Demo Program (PULSIN.bs1)

```
' PULSIN.bs1
' This program uses PULSIN to measure a pulse generated by discharging a
' 0.1 uF capacitor through a 1k resistor. Pressing the switch generates
' the pulse, which should ideally be approximately 120 us (12 PULSIN units
' of 10 us) long. Variations in component values may produce results that
' are up to 10 units off from this value. For more information on
' calculating resistor-capacitor timing, see the RCTIME command.

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

SYMBOL  Pulse           = 7             ' pulse input pin

SYMBOL  time            = W1            ' pulse width (10 uS units)
```

```
Main:
  PULSIN Pulse, 1, time                 ' measure positive pulse
  IF time = 0 THEN Main                 ' if 0, try again
  DEBUG CLS, time                       '   else display result
  GOTO Main
  END
```

All 2

Demo Program (PULSIN.bs2)

NOTE: This example program can be used with all BS2 models. This program uses conditional compilation techniques; see Chapter 3 for more information.

```
' PULSIN.bs2
' This program uses PULSIN to measure a pulse generated by discharging a
' 0.1 uF capacitor through a 1K resistor. Pressing the switch generates
' the pulse, which should ideally be approximately 120 us (60 PULSIN units
' of 2 us) long (for BS2 and BS2e). Variations in component values may
' produce results that are up to 10 units off from this value.  For more
' information on calculating resistor-capacitor timing, see the RCTIME
' command.

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

Pulse           PIN     7               ' pulse input pin

#SELECT $STAMP
  #CASE BS2, BS2E, BS2PE
    Scale       CON     $200            ' 2.0 us per unit
  #CASE BS2SX, BS2P
    Scale       CON     $0CC            ' 0.8 us per unit
  #CASE BS2PX
    Scale       CON     $0CF            ' 0.81 us per unit

#ENDSELECT

time            VAR     Word

Main:
  PULSIN Pulse, 1, time                 ' measure positive pulse
  IF (time > 0) THEN                    ' if not 0
    DEBUG HOME,
          DEC time, " units ", CLREOL   ' display raw input
    time = time */ Scale                ' adjust for Stamp
    DEBUG CR,
          DEC time, " us  "             ' display microseconds
  ELSE
    DEBUG CLS, "Out of Range"           ' else error message
  ENDIF
  PAUSE 200
  GOTO Main
  END
```