RCTIME

BS1	BS2	BS2e	BS2sx	BS2p	BS2pe	BS2px

1 **(See POT)**

All 2 **RCTIME *Pin, State, Variable***

Function

Measure time while *Pin* remains in *State*; usually to measure the charge/discharge time of resistor/capacitor (RC) circuit.

- ***Pin*** is a variable/constant/expression (0 – 15) that specifies the I/O pin to use. This pin will be placed into input mode.
- ***State*** is a variable/constant/expression (0 - 1) that specifies the desired state to measure. Once *Pin* is not in *State*, the command ends and stores the result in *Variable*.
- ***Variable*** is a variable (usually a word) in which the time measurement will be stored. The unit of time for *Variable* is described in Table 5.87.

Quick Facts

Table 5.87: RCTIME Quick Facts.

	BS2	BS2e	BS2sx	BS2p	BS2pe	BS2px
Units in *Variable*	2 µs	2 µs	0.8 µs	0.75 µs	2 µs	0.75 µs
Maximum Pulse Width	131.07 ms	131.07 ms	52.428 ms	49.151 ms	131.07 ms	49.151 ms

Explanation

RCTIME can be used to measure the charge or discharge time of a resistor/capacitor circuit. This allows you to measure resistance or capacitance; use R or C sensors such as thermistors or capacitive humidity sensors or respond to user input through a potentiometer. In a broader sense, RCTIME can also serve as a fast, precise stopwatch for events of very short duration.

HOW RCTIME'S TIMER WORKS.

When RCTIME executes, it makes *Pin* an input, then starts a counter (who's unit of time is shown in Table 5.87). It stops this counter as soon as the specified pin is no longer in *State* (0 or 1). If pin is not in *State* when the instruction executes, RCTIME will return 1 in *Variable*, since the instruction requires one timing cycle to discover this fact. If pin remains in *State* longer than 65535 timing cycles RCTIME returns 0.

Figure 5.33 shows suitable RC circuits for use with RCTIME. The circuit in Figure 5.33a is preferred, because the BASIC Stamp's logic threshold is approximately 1.4 volts. This means that the voltage seen by the pin will start at 5V then fall to 1.4V (a span of 3.6V) before RCTIME stops. With the circuit of Figure 5.33b, the voltage will start at 0V and rise to 1.4V (spanning only 1.4V) before RCTIME stops. For the same combination of R and C, the circuit shown in Figure 5.33a will yield a higher count, and therefore more resolution than Figure 5.33b.

SUITABLE RCTIME CIRCUITS.

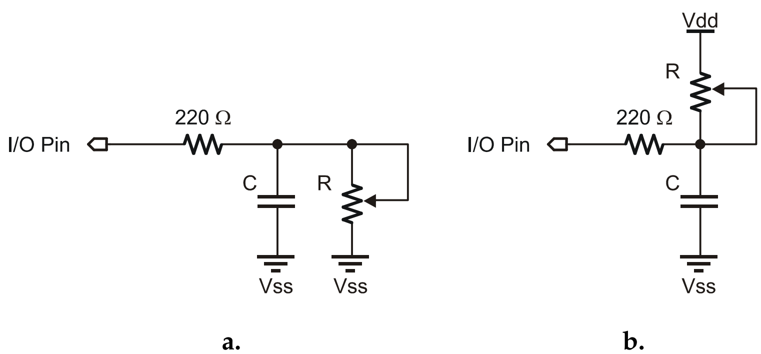


Figure 5.33: Example RC Circuits. Use **a** (left) with *State* = 1. Use **b** (right) with *State* = 0.

Before RCTIME executes, the capacitor must be put into the state specified in the RCTIME instruction. For example, with Figure 5.33a, the capacitor must be charged until the top plate is at 5V, then a State value of 1 will be used to monitor the discharge of the capacitor through the variable resistance.

DON'T FORGET TO DISCHARGE THE CAPACITOR BEFORE EXECUTING RCTIME.

Here's a typical sequence of instructions for Figure 5.33a (assuming I/O pin 7 is used):

```
result  VAR     Word

HIGH 7                                  ' charge the cap
PAUSE 1                                 '   for 1 ms
RCTIME 7, 1, result                     ' measure RC discharge time
DEBUG DEC ? result                      ' display result
```

Using RCTIME is very straightforward, except for one detail: For a given R and C, what value will RCTIME return? It's easy to figure, based on a value called the RC time constant, or tau (τ) for short. Tau represents the time required for a given RC combination to charge or discharge by 63

PREDICTING THE RETURNED VALUE.

percent of the total change in voltage that they will undergo. More importantly, the value τ is used in the generalized RC timing calculation. Tau's formula is just R multiplied by C:

$\tau = R \times C$

CALCULATING CHARGE AND DISCHARGE TIME.

The general RC timing formula uses τ to tell us the time required for an RC circuit to change from one voltage to another:

$time = -\tau * (\ln (V_{final} / V_{initial}))$

In this formula ln is the natural logarithm; it's a key on most scientific calculators. Let's do some math. Assume we're interested in a 10 k resistor and 0.1 µF cap. Calculate τ:

$\tau = (10 \times 10^{3}) \times (0.1 \times 10^{-6}) = 1 \times 10^{-3}$

The RC time constant is 1×10^{-3} or 1 millisecond. Now calculate the time required for this RC circuit to go from 5V to 1.4V (as in Figure 5.33a):

$time = -1 \times 10^{-3} \times (\ln(1.4v / 5.0v)) = 1.273 \times 10^{-3}$

THE RC TIME EQUATION.

On the BS2, the unit of time is 2µs (See Table 5.87), that time (1.273×10^{-3}) works out to 636 units. With a 10 kΩ resistor and 0.1 µF cap, RCTIME would return a value of approximately 635. Since $V_{initial}$ and V_{final} doesn't change, we can use a simplified rule of thumb to ***estimate*** RCTIME results for circuits like Figure 5.33a:

RCTIME units = 635 x R (in kΩ) x C (in µF)

DETERMINING HOW LONG TO CHARGE OR DISCHARGE THE CAPACITOR BEFORE EXECUTING RCTIME.

Another handy rule of thumb can help you calculate how long to charge/discharge the capacitor before RCTIME. In the example above that's the purpose of the HIGH and PAUSE commands. A given RC charges or discharges 98 percent of the way in 5 time constants (5 x R x C). In Figure 5.33, the charge/discharge current passes through the 220 Ω series resistor and the capacitor. So if the capacitor were 0.1 µF, the minimum charge/discharge time should be:

$\text{Charge time} = 5 \times 220 \times (0.1 \times 10^{-6}) = 110 \times 10^{-6}$

So it takes only 110 µs for the cap to charge/discharge, meaning that the 1 ms charge/discharge time of the example is plenty.

NOTES ABOUT 220 Ω RESISTOR IN THE RC CIRCUITS.

A final note about Figure 5.33: You may be wondering why the 220 Ω resistor is necessary at all. Consider what would happen if resistor R in Figure 5.33a were a pot, and were adjusted to 0 Ω. When the I/O pin went high to charge the cap, it would see a short direct to ground. The 220 Ω series resistor would limit the short circuit current to 5V/220 Ω = 23 mA and protect the BASIC Stamp from damage. (Actual current would be quite a bit less due to internal resistance of the pin's output driver, but you get the idea.)

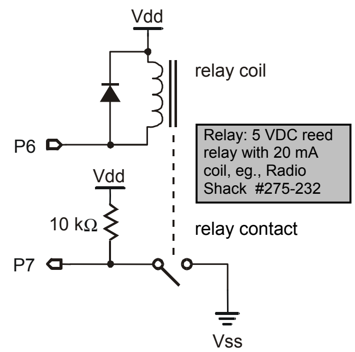
Demo Program (RCTIME1.bs2)

All 2

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

```
' RCTIME1.BS2
' This program shows the standard use of the RCTIME instruction measuring
' an RC charge/discharge time. Use the circuit in the RCTIME description
' (in the manual) with R = 10K pot and C = 0.1 uF. Connect the circuit to
' pin 7 and run the program.  Adjust the pot and watch the value shown on
' the Debug screen change.

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

RC              PIN     7

result          VAR     Word


Main:
  DO
    HIGH RC                             ' charge the cap
    PAUSE 1                             '   for 1 ms
    RCTIME RC, 1, result                ' measure RC discharge time
    DEBUG HOME, DEC result              ' display value
    PAUSE 50
  LOOP
  END
```

Figure 5.34: Relay Circuit for Demo Program RCTIME2.bs2.



All 2

Demo Program (RCTIME2.bs2)

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

```
' RCTIME2.BS2
' This program illustrates the use of RCTIME as a fast stopwatch. The
' program energizes a relay coil, then measures how long it takes for the
' relay contacts to close.  The circuit for this program can be found in
' the manual. Note that RCTIME doesn't start timing instantly -- as with
' all PBASIC instructions, it must be fetched from program EEPROM before
' it can execute.

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

Coil            PIN     6
RC              PIN     7

#SELECT $STAMP
  #CASE BS2, BS2E, BS2PE
    Adjust      CON     $200            ' x 2 us per unit
  #CASE BS2SX
    Adjust      CON     $0CC            ' x 0.8 us per unit
  #CASE BS2P, BS2PX
    Adjust      CON     $0C0            ' x 0.75 us per unit
#ENDSELECT

result          VAR     Word


Main:
  DO
    LOW Coil                            ' energize relay coil
    RCTIME RC, 1, result                ' measure time to contact closure
    result = result */ Adjust           ' adjust for device
    DEBUG "Time to close: ",
```

```
        DEC Result, CR
  HIGH Coil                           ' release relay
  PAUSE 1000                          ' wait one second
LOOP
END
```