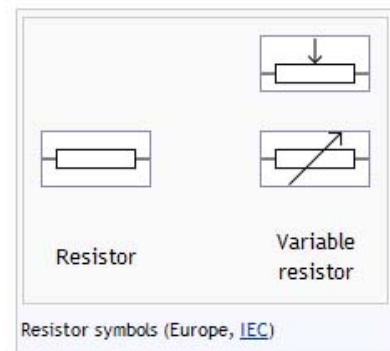
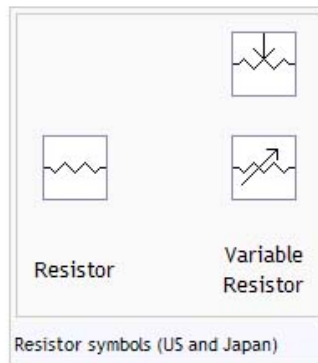
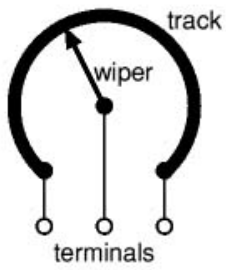


```
'This is demo before chapter 5
' {$STAMP BS2}
' {$PBASIC 2.5}
'Program demonstrating use of variables,
'doing arithmetic, conditional looping
  A VAR Word
  B VAR Word
  Total VAR Word

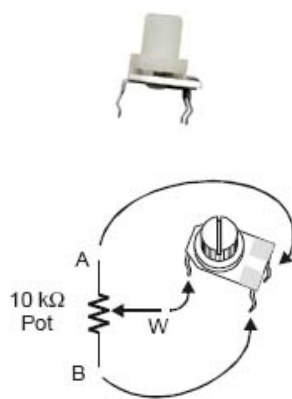
DO
  DEBUG CLS,"What is the first number?",CR
  DEBUGIN DEC a
  DEBUG "What is the second number?",CR
  DEBUGIN DEC B
  Total = a+b
  DEBUG "The total is ",DEC Total
  DEBUG CR, "Want to try again? (n=no y=yes)",CR
  DEBUGIN a
LOOP WHILE (a = "y")
  DEBUG CR,"Farewell"
END
```

## Chapter 5 Measuring Rotation

**Potentiometer (Pot): A variable resistor.**

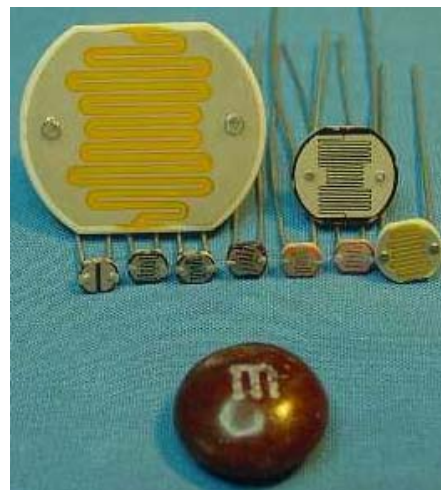


## Breadboard Variable Resistor



**Figure 5-3**  
Potentiometer  
Schematic Symbol  
and Part Drawing

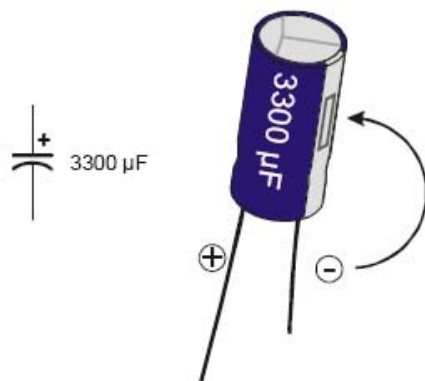
## Photoresistors



**You can measure resistance by measuring time!**

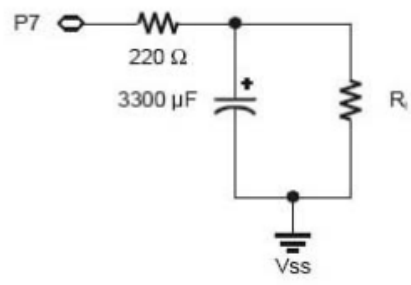
**How?**

**Connect a resistor and capacitor in circuit. Measure the amount of time it takes to charge or discharge the capacitor. The longer the time, the larger the resistance.**



**Figure 5-7**  
3300 µF Capacitor  
Schematic Symbol  
and Part Drawing

*Pay careful attention  
to the leads and how  
they connect to the  
Positive and  
Negative Terminals.*



```
' What's a Microcontroller - PolledRcTimer.bs2
' Reaction timer program modified to track an RC-time voltage decay.

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

```
timeCounter  VAR    Word
counter      VAR    Nib
```

```
DEBUG CLS
```

```
HIGH 7
DEBUG "Capacitor Charging...", CR
```

```
FOR counter = 5 TO 0
  PAUSE 1000
  DEBUG DEC2 counter, CR, CRSRUP
NEXT
```

```
DEBUG CR, CR, "Measure decay time now!", CR, CR
INPUT 7
```

```
DO
```

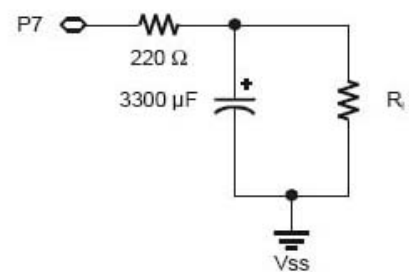
```
  PAUSE 100
  timeCounter = timeCounter + 1
```

```
  DEBUG ? IN7
  DEBUG DEC5 timeCounter, CR, CRSRUP, CRSRUP
```

```
LOOP UNTIL IN7 = 0
```

```
DEBUG CR, CR, CR, "The RC decay time was ",
  DEC timeCounter, CR,
  "tenths of a second.", CR, CR
```

```
END
```



```

' What's a Microcontroller - ReadPotWithRcTime.bs2
' Read potentiometer in RC-time circuit using RCTIME

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

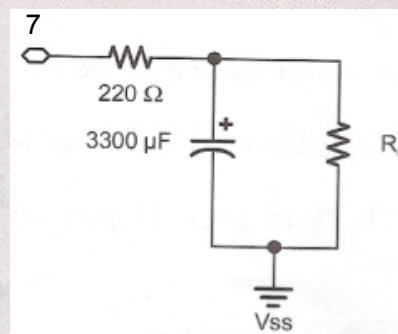
time VAR Word

DO

HIGH 7
PAUSE 100
RCTIME 7, 1, time
DEBUG HOME, "time = ", DEC5 time

LOOP

```



Syntax: **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 below.

	BS2, BS2e, and BS2pe	BS2sx	BS2p	BS2px
Units in <i>Variable</i>	2 $\mu$ s	0.8 $\mu$ s	0.75 $\mu$ s	0.75 $\mu$ s
Maximum pulse width	131.07 ms	52.428 ms	49.151 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.

When **RCTIME** executes, it starts a counter (whose unit of time is shown above). 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.