

## Instructions

The F-15E Strike Eagle is the top air-superiority fighter in the world, even though its design is 25 to 35 years old. It can carry up to eight AIM-120 AMRAAM missiles at a time for use in dogfights.



You must design a circuit based on the BASIC Stamp to set and monitor the onboard missile complement. The program must be demonstrated without being attached to the Board of Education; however, to avoid accidents, do as much development as possible with the Stamp on the BoE. This project can use as many as all 16 pins of the Stamp, as well as one D flip flop on the momentary switch to be sure your program doesn't miss a signal. You are free to use additional circuit components in your own design.

- Input Summary:** One potentiometer, generating numbers from 1 through 6  
One SPDT switch "mode"  
One momentary switch "load" or "fire" (depending on mode)
- Output Summary:** Three seven-segment displays:  
a. Number of missiles to load  
b. Number of missiles onboard  
c. Number of missiles to fire  
One warning LED: invalid load/fire request  
One confirmation tone speaker

The aircraft will be in one of two modes: *grounded* or *combat*, as set by an SPDT switch. Regardless of the mode, the total number of missiles onboard is indicated on the second display, and can never exceed 8.

While the aircraft is *grounded*, firing is disabled, so the third display is blank (not zero!). New missiles can be loaded onto the aircraft. The number of missiles to be loaded at one time can be varied from 1 to 6 by turning the potentiometer; this value is continually displayed/updated on the first display. The loading action is triggered by the momentary switch: the second display increments towards its new total in steps lasting 0.5 seconds each, during which time, new "load" commands are ignored. If a load is requested that would put the onboard total over 8, no missiles are loaded, and instead the warning light comes on for exactly one second.

When the aircraft is in *combat* mode, the first display is blank. The number of missiles to be fired is now read from the potentiometer (again, ranging 1 to 6) and continually shown in the third display. The firing action is triggered by the same momentary switch: the second display decrements towards its new total in steps lasting 0.5 seconds each, during which time, new "fire" commands are ignored. As each missile is fired, a confirmation tone sounds for 0.25 seconds. If a fire command requests more missiles than available, then none are fired, and the warning light comes on for exactly one second.

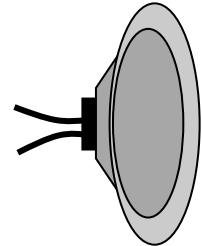
For fun, you may also have a loading confirmation tone. If so, it must be a different frequency.

## Analog Inputs and Outputs

This project requires some analog inputs and outputs that would be very difficult to implement using TTL digital circuits, but are relatively easy with the Stamp.

### Speakers

A speaker is an output device that essentially translates voltage variations into the motion of a small membrane. This membrane pushes on the nearby air, which generates sound. People can hear sounds ranging roughly from 20 Hz to 20,000 Hz. Most speakers act electrically as resistors of very small resistance (usually less than  $10\Omega$ ), which makes them difficult to control with digital outputs. Further, digital waveforms make harsh, buzzy sounds.

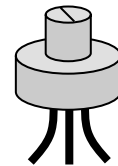


PBASIC has a command called **FREQOUT pin, duration, frequency** which can be used to output a smoothly oscillating voltage, a decidedly non-digital operation. One wire of a speaker can be connected to such an output, with the other wire connected to ground. For example, the following command generates a sound of frequency 200 Hz for one and a half seconds on pin 4:

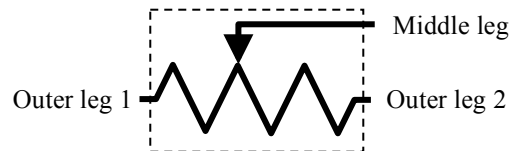
```
FREQOUT 4, 1500, 200
```

### Potentiometers

A potentiometer, or “pot” for short, is a resistor-like circuit element having a variety of uses. We are interested in using one as a variable resistor, and then reading that variable value using the Stamp.



As suggested by the schematic for a potentiometer, it has 3 wires, with a simple resistor between the outer two wires. This resistor has a constant value,  $R_{max}$ . The middle wire is connected to a “wiper,” which makes contact on the resistor somewhere between the two ends, as controlled by a knob. As the wiper approaches one end, the resistance between that end and the middle wire drops towards zero, while the resistance between the middle wire and the other end increases towards  $R_{max}$ .

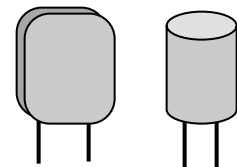


Terminal 1 to 3	Terminal 1 to 2	Terminal 2 to 3
1000 $\Omega$	0 $\Omega$	1000 $\Omega$
1000 $\Omega$	200 $\Omega$	800 $\Omega$
1000 $\Omega$	500 $\Omega$	500 $\Omega$
1000 $\Omega$	800 $\Omega$	200 $\Omega$
1000 $\Omega$	1000 $\Omega$	0 $\Omega$

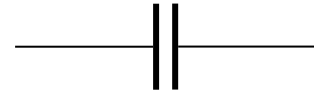
Thus, if we connect to only the middle wire and one end, the pot acts as a variable resistor.

### Capacitors

A capacitor is a device that stores charge. It has two legs, like a resistor. As current flows into it, charge accumulates like water in a swimming pool. However, the more it fills up, new charge is accepted into it more slowly. Eventually, the capacitor will become completely filled up, unless something is



done to empty it. Once it is full, no new charge can enter, and therefore, no current can flow along any wires going into it. The units of capacitance are Farads, named after Michael Faraday. A one Farad capacitor is huge; you will probably not see one in your life. The symbol for a capacitor is two parallel lines:

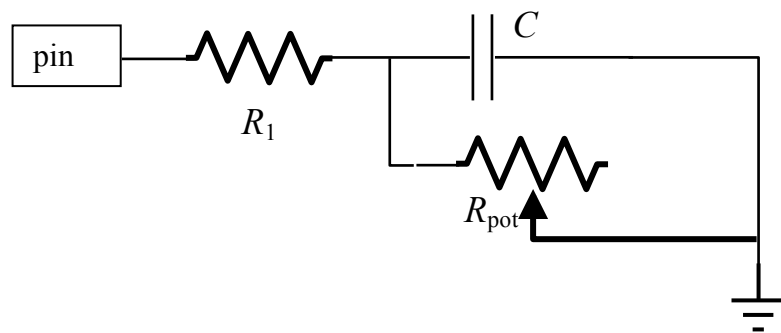


## Time Constants and RCTIME

The Stamp cannot measure resistance, nor can it measure capacitance, but it can measure time (because it has an internal clock). In particular, it has a command that can be used to measure how long it takes to fill a capacitor by running current through a resistor. This will depend on how big the capacitor is (bigger  $\rightarrow$  longer to fill), and also on the resistance of the resistor (bigger  $\rightarrow$  less current  $\rightarrow$  longer to fill). The Stamp can also measure how long it takes to empty a capacitor.

For any given resistor  $R$  and capacitor  $C$ , a “typical” fill or emptying time is given by the product  $\tau = RC$ , which is called the “time constant” of the combination. The units work out to be Ohms\*Farads=seconds, rather surprisingly! In that time, a capacitor will “charge up” from empty to about 63% full, or “discharge” from 100% full to about 37% full. But those exact numbers aren’t too important. What’s important is that for **any** specific amount of filling or draining (say, from 95% full to 22% full), the amount of time required is proportional to  $RC$ .

Consider the following circuit, connecting a capacitor and some resistors to a pin on the Stamp:



- First, we’ll set the pin high (+5V) for some time, long enough that we are sure the capacitor is full. Waiting  $5\tau_1 = 5R_1C$  should be enough. (Choose  $R_1 \geq 220\Omega$  to keep from demanding too much current from the Stamp.) Note that during this time, current is also flowing through  $R_{pot}$  to ground. This means that we need  $R_{pot} \gg R_1$ , so that we can fill the capacitor faster than we are draining it.
- Then, we’ll disconnect the pin from its latch inside the STAMP. Note we are **not** setting it to zero, merely disconnecting it. The charge stored in the capacitor will start to empty through  $R_{pot}$  to ground.
- As the capacitor empties, the voltage on the pin will slowly drop. The Stamp will repeatedly monitor the status of this input until it sees the voltage as a low signal (about 1.4V, actually). The time required will be proportional to the resistance  $R_{pot}$ . If the knob is turned all the way to one side so that  $R_{pot}$  is zero, then it will take roughly  $\tau_{min} = 0$ . If the knob is turned all the way to the other side, then it will take roughly  $\tau_{max} = R_{max}C$ .

A single PBasic command accomplishes steps (b) and (c): **RCTIME pin, 1, varname**. The 1 means to monitor a draining capacitor (0 can be used for filling capacitors). During step (c), the Stamp checks the voltage about 500,000 times per second (for version 2 of the Stamp); when the process is done, the number of times the voltage was checked is left in the variable **varname**.

## Implementation

Get a program running that can successfully monitor a potentiometer before proceeding with the rest of the project. Some advice:

- a. Determine  $R_{\max}$  for the pot. If you have  $R_{\max} < 10R_1 = 2200\Omega$  then you have a problem; see the instructor for an alternate (and not as good) way to wire your circuit.
- b. You want RCTIME to be able to count up to a moderately large number (let's say 1000) when the pot is at its max. Thus you want a time constant

$$\frac{1000}{500,000 / \text{s}} = 2 \text{ ms} = \tau_{RC} = R_{\max} C . \text{ Choose a capacitor near } C \approx \frac{2 \text{ ms}}{R_{\max}} .$$

- c. These are only guidelines. Resistors and, especially, capacitors are only available in certain sizes. Do the best you can!
- d. You can recalculate the RCTIME counts that you expect for your specific resistors and capacitors. However, they are unlikely to be correct. Individual resistors and, especially, capacitors can have values significantly different from their “nominal” value — up to 10%, sometimes 20%, off. Instead, write a program to measure what counts you get from various knob settings.