

Hello, Photon!:

Objective: Help you work on your coding/problem-solving skills; encourage you to read our C++ textbook, notes; and independently research Particle's API.

Instructions: Write each of the 4 programs below. Each will
a) be an independent Particle Workbench project and
b) utilize a function `flashD7()`

flashD7():

This function will have two double parameters, **onTime** and **offTime**. It will cause D7 to turn on for onTime seconds, and then off for offTime seconds. It will do this just once (not in a loop). You will use `flashD7()` in the programs 0-2.

Program 0:

Write a program that will flash D7 on for a random amount of time between 0.5-1.5 seconds, and then off for 0.5 seconds. You will find Particle's random number capabilities particularly useful.

Program 1:

Write a program that will flash D7 on for 100, 200, 300, 400ms, 500ms (always off for 500ms in between) and then repeat.

Program 2:

Write a program that will flash D7 on for $2.0 * a + 100.0$ (always off for 500 ms in between) where **a** will take on each of the ASCII codes in your first name -- and then **repeat**. For example, if your name was CAZ (67, 65, 90), you would flash it on for { $2 * 67 + 100$, $2 * 65 + 100$, $2 * 90 + 100$ } ms.

Program 3:

Read the API on **RGB** and have your Photon *do something visually exciting, and programmatically non-trivial (e.g., not just 1 or 2 lines of code)* with it

Loops: All 4 of these programs involve doing something a specific number of times. This would suggest setting up a for-loop inside of `loop()`, but if the for-loop executes for too long (e.g., due to programmer error), the Photon will lose its connection to the Particle cloud, necessitating you place it in safe mode.

There are at least 3 ways to solve this problem. One is to make sure that the loops are very short -- probably that will work in this case. A second is to call **`Particle.process()`**, within the loop, to ping the cloud and keep the Photon connected. A third is to embed the body of the for-loop inside an if statement, and use an external, global variable to control the number of repetitions. Any code that has to execute after the loop would be placed in the else part of the if.

Potentially Fraught Strategy	Preferred Strategy
<pre> void setup(){ } void loop(){ for(int count=0; count < 100; count++){ // flash an LED in some fashion } afterTheLoop(); } </pre>	<pre> int count; void setup(){ count = 0; } void loop(){ if(count < 100){ // flash an LED in some fashion count++; } else { afterTheLoop(); count = 0; } } </pre>