# Local Skills - HelloWorldpart3_TimerEvent

In Part 3 of *Local Skills - Hello World* we'll use timed events to trigger a change in Misty's chest LED every second. Timed events allow us to specify an amount of time to pass before an event occurs and the callback is triggered. In addition, we'll introduce global variables and demonstrate how they persist across new threads.

## HelloWorldpart3_TimerEvent.json

```
{
      "Name": "HelloWorldpart3_TimerEvent",
      "UniqueId": "8a380289-2939-4e81-94d9-86d511b7a8ce",
      "Description": "Local 'Hello, World!' tutorial series, part 3.",
      "StartupRules": [ "Manual", "Robot" ],
      "Language": "javascript",
      "BroadcastMode": "verbose",
      "TimeoutInSeconds": 300,
      "CleanupOnCancel": false,
      "WriteToLog": false
}
```

## HelloWorldpart3_TimerEvent.js

When registering for a timed event use the `RegisterTimerEvent()` method. Pass in the name of the event we're creating, the amount of time (in ms) we want Misty to wait before triggering the callback function, and set the `keepAlive` parameter to `true` in order to have the event trigger the callback automatically every 3 seconds until it is unregistered. After the line of code to register for the timer event, send our first command to change Misty's LED to white below the timer event. This will turn the LED on for the first 3 seconds our skill runs, before the first callback is fired.

```
misty.RegisterTimerEvent("TimerEvent", 3000, true);
misty.ChangeLED(255, 255, 255); // white
```

Define a global variable to track the amount of callbacks that have been triggered. In order for the data to persist across new threads created by callbacks, prefix the name with an underscore. Initialize the value of the variable as "0". Declare it **above** the `RegisterTimerEvent()` method. **Note:** Do not include a type when creating global variables.

```
_count = 0;
```

In the callback function (automatically named `_TimerEvent`) start by checking if `_count` is less than `5` using an `if..then` statement. If so, increment `_count` by one to keep track of the amount of times we are changing the LED. Then, generate three random values between 0 and 255 and pass them in to `ChangeLED` to trigger a change in Misty's chest LED.

```
if (_count < 5) {
   _count = _count + 1;

   let value1 = Math.floor(Math.random() * (256));
   let value2 = Math.floor(Math.random() * (256));
   let value3 = Math.floor(Math.random() * (256));
   misty.ChangeLED(value1, value2, value3);
} else {
}
```

The `else` statement will trigger once `_count` has reached `5`. At this point, we want the skill to end. Start by unregistering for the timer event by calling `UnregisterEvent` and passing in the name designated for the event. Then turn the LED off by passing in zero values for `ChangeLED` and log a debug message.

```
else {
   misty.UnregisterEvent("TimerEvent");
   misty.ChangeLED(0, 0, 0); // off
   misty.Debug("ending skill helloworld part3");
}
```

Using timed events we have told Misty to change her chest LED to a random color in three-second intervals. We've demonstrated how we can use global variables prefixed with an underscore to have data persist across threads that are created in our program as callbacks are triggered. This is a simple example of two powerful tools that you have at your disposal when writing local skills for Misty. See the skill file below for reference.

```
// debug message to indicate the skill has started
misty.Debug("starting skill helloworld part3");

// global variable to track the amount of callbacks triggered
_count = 0;
```

```
// register for the timer event, specifying the duration of the timer
misty.RegisterTimerEvent("TimerEvent", 3000, true);

// callback specified for Timer event
function _TimerEvent() {
    // check if _count is less than 5
    if (_count < 5) {
        // increment count by 1
        _count = _count + 1;

        // specify random RGB values and send command to change LED
        let value1 = Math.floor(Math.random() * (256));
        let value2 = Math.floor(Math.random() * (256));
        let value3 = Math.floor(Math.random() * (256));
        misty.ChangeLED(value1, value2, value3);
    } else {
        // otherwise, turn off LED, unregister for the timer event and
        // signal end of skill
        misty.UnregisterEvent("TimerEvent");
        misty.ChangeLED(0, 0, 0); // off
        misty.Debug("ending skill helloworld part3");
    }
}
```