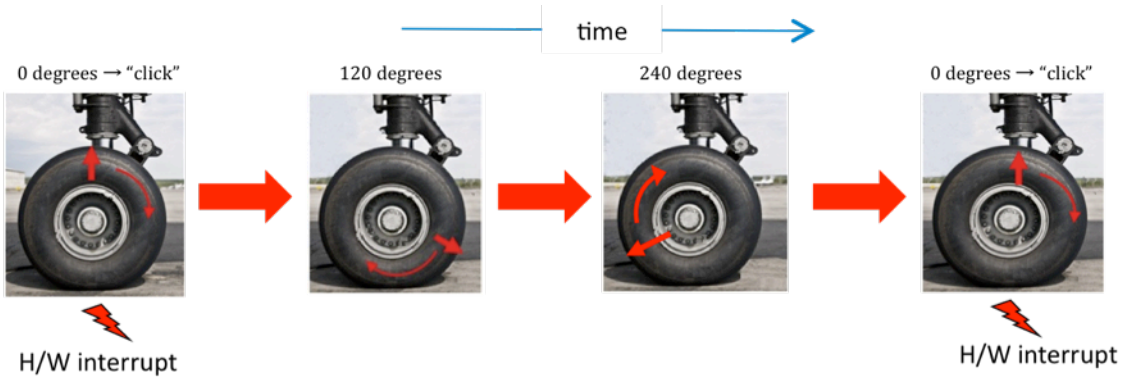
Nose Gear (NG) Velocity Example

Version 1.1, September 2011
Critical Systems Labs Inc.

This is a completely fictitious example of a simple function that estimates the velocity of an aircraft while moving on the ground.

Overview: The velocity is estimated by measuring the elapsed time of a full rotation of the nose gear wheel. Each time this wheel completes a full rotation, a pulse (informally called a "click") is generated by an electro-mechanical sensor connected to a computer. The pulse causes a hardware interrupt which is serviced by an interrupt routine that increments a 16-bit counter called "NGRotations" and stores the current time in a 16-bit variable called "NGClickTime". The Real-time Operating System (RTOS) periodically invokes another function that uses the current value of this counter to re-calculate an estimate of the current velocity and store the result in a global variable called "estimatedGroundVelocity". The RTOS ensures that this update function is invoked at least once every 500 milliseconds, however, the exact timing of each invocation relative to a hardware interrupt is not predictable. In addition to the counter that records rotations of the wheel, this update function has read-only access to a 16-bit counter called "Millisecs" which is incremented once every millisecond. This counter is the same source of time used by the interrupt routine to update NGClickTime. The circumference of the nose gear wheel is also available to the update function as the value of a compile-time constant called "WHEEL_DIAMETER". The update function may store private data values that are protected from invocation to invocation. An example of this calculation is shown below:



WHEEL_DIAMETER = 22 inches
PI = 3.14

12 inches/foot
5280 feet/mile

estimatedGroundVelocity = distance travel/elapsed time
= ((3.14 * 22)/(12*5280))/((4367-4123)/(1000*3600))
= 16 mph

In this particular example, it is assumed that NGRotations has just been incremented when the calculation is performed, i.e., the update function is invoked within a few milliseconds after a new 'click'. However, the implementation of this function should handle the more general case in which the update function can be invoked at any time.

Requirement 1: When the estimatedGroundVelocityIsAvailable flag is true (i.e., while the aircraft is moving on the ground an this global variable is set to a non-zero value), the value of the global variable estimatedGroundVelocity shall be within 3 km/hr of the true velocity of the aircraft at some moment within the past 3 seconds.

This requirement may be expressed in a semi-formal manner as follows (where behaviour is modelled at the millisecond time scale):

```
FORALL t,
  IF (estimatedGroundVelocityIsAvailable at time (t + 3000))
  THEN EXISTS n, m SUCH THAT:
   ((1 <= n) AND (n <= 3000)) AND
   ((-3 <= m) AND (m <= 3)) AND
   (estimatedGroundVelocity at time (t + 3000) =
     (trueGroundVelocity at time (t + n)) + m)
```

Assurance Case(s): The assurance cases must provide evidence that the design (or implementation) of the update function satisfies the required behaviour. Note that the above statement of the required behaviour is expressed in terms of a relationship between a physical object (i.e., the aircraft) and the value of a variable in the memory of the computer. All assumptions needed to support the assurance cases must be explicitly stated. The evidence provided by these assurance cases must be at least as thorough as the evidence that would required for RTCA DO 178B certification of Level A software including MCDC test coverage. (If formal analysis is used instead of testing, then the assurance cases must be accompanied by an argument that the results of formal analysis are at least as thorough as what would be achieved by MCDC test coverage.)

