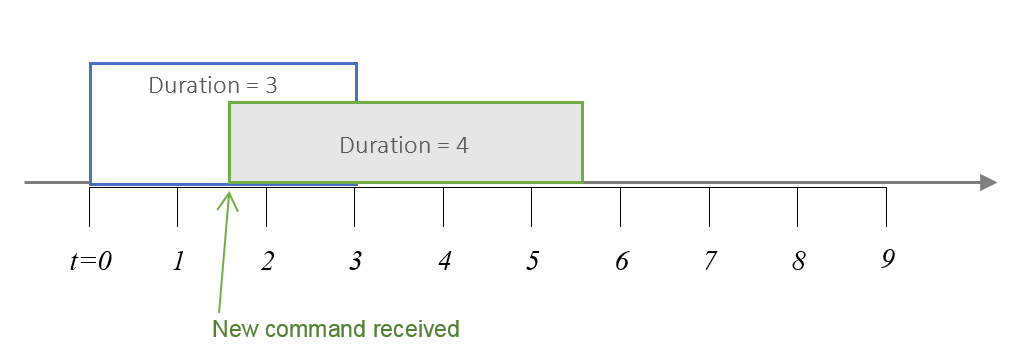
# Objective: Build a basic digital acquisition board accompanied by a PC application.

# Design

This project consists of several core components.

1. Protocol – command for the microcontroller. Response from the microcontroller. Document on how the command protocol works.
2. Encoder – C# code to produce command
3. Parser – C code to extract command content so that the microcontroller can act on the command.
4. Driver
   1. Periodic waveform - 50% duty cycle, user inputs pulse width.
   2. Timed output – user inputs duration (in number of seconds). Timing starts when user enables the output. When the output pin’s active duration equals or exceeds the duration, the output pin reverts to default state (off state).



* 1. ADC – return raw ADC value on input pin. Developer decides and documents which sampling method is being used. Sampling methods can be:
     1. On-demand – sample when command is received.
     2. Background – sample periodically in the background. Returns the most recent ADC value when command is received.

***Instructor’s comment:***

*“A good design should be extensible. A good implementation can also withstand slight changes in design. However, design and implementation are iterative processes. There is no such thing as perfect design. Try to bring your idea to a working state and then refactor your implementation up to a point you feel complete rewrite would be a better-off option.”*

# Implementation

1. Implementation should be done on PIC24FJ256GA702
2. Implementation should use APIs provided by FreeRTOS
   1. Use co-routine for heartbeat with interval of 1 second
   2. Use task for UART data transceiving.
   3. Use software timer, vTaskDelay(), for delay.
3. Document flowchart on encoder and parser.
4. Document components dependencies.
5. Document limitations on user input (such as limiting user input to integer only).