Design Challenge: Digipot control driver

# Guidelines

The problem below is for you to work through in your own time. Write up your solution and email before your design challenge review. Please do not spend more than 10 hours on this design – and keep in mind you may be able to spend significantly less than that and still effectively demonstrate your engineering skill.

Feel free to make reasonable assumptions, since we haven’t specified every detail of the hypothetical embedded target platform for this design. If the requirements are unclear or insufficient, you are encouraged to contact us with questions.

# Overview

Write a C module to provide an application means to control a dual-channel digital potentiometer (MAX5389, 10 kΩ model)

# Requirements

* API
  + Provides ability to set the Low Terminal to Wiper Terminal resistance, for each channel independently, to a specified resistance
  + Provides feedback as to whether the resistance commanded has been achieved
* Resistance setting must take < 40ms
* Code in C language
* Only libraries allowed are the C standard library – all code provided must be your own

# Assume

* 132 MHz processor clock rate
* stdout and stdin are already setup to access serial port available for debug
* No operating system
* Other processes may be running in target applications as well, including (for example)
  + 20 kHz loop for power electronics feedback control
  + 2 kHz critical communication loop
  + Low speed serial port communication

# Provided

* “Periodic.h” - Header file for module that provides services to configure a programmable timer with a specified frequency and interrupt handler function, start/stop timer, enable/disable interrupts
* “Pin.h” - Header file for module that provides ability to write to pin, and defines pin names
* “Generic.h” - Header file including standard typedefs for scalar types and etc
* “MAX5389.pdf” - Datasheet for MAX5389
  + In “TIMING CHARACTERISTICS” on page 3, change “MHz” to “kHz” and “ns” to “μs” for the sake of this challenge

# Deliverables

* High-level design, including block diagrams, test cases, textual descriptions, flow diagrams, or anything you feel is helpful
* Module .h and .c files
* main function in separate file to exercise module’s functionality