Documentation:

USES: mbed OS PWM driver, uses PwmOut class.
Controls microcontroller pin for PWM.

- Hardware driven, so does not require CPU clock cycles.

Duty cycle, such as in PwmOut.write(0.75f), means the Pwm is ON 75% of each interval, measured in frequency (Hz).

mbed::Ticker sampleTicker;

- Controls rate for precise timing calculation. Switching tables samples at a hardware controlled rate. Updating exactly ever 66.6 us. This counteracts JITTER
    - Jitter: variation in timing when an event should happen vs when it happens.

DDS Algorithm: to generate smooth sine with math, not hardware

- ☐ Think of it like a digital music box:
    - The lookup table(music roll with notes), phase accumulator(mechanism moving roll), ISR(Person pushing the roll).
- Phase accumulator: circular, phase at 1 returns to 0.
    - If a lookup table has 100 samples, phase will increment by 0.01. We can change speed by changing the increment.
        - Phase += 0.05 would take 200 samples before a complete loop
- Lookup Table
    - Bypasses calculating sin every time which is slow. Instead we look it up from the lookup table, pre calculated sine values, taking inputs as predetermined table sizes for calculations.
- ISR: interrupt service Routine
    - ISR is a hardware level event, Each defined cycle, it interrupts CPU and and current calculations in Loop and performs all in OnSampleTicker():

LINEAR INTERPOLATION

- If we only have 100 table entries, but our phase is 0.015, we guess the value between two table entries. IE: guess between index 0.01 and 0.02.

A Complete View:

Through a combination of a lookup table, phase accumulator, and ISR, the program runs as follows: A fixed-size lookup table containing one complete cycle of sine wave values scaled 0-1 is pre-calculated once at startup. The user's desired frequency determines a phase increment value, which controls how rapidly the phase accumulator advances through the table on each ISR call. A hardware timer triggers the ISR at precise, fixed intervals (e.g., every 100μs for 10kHz sample rate), and during each ISR execution, the current phase value is used to look up or interpolate between table entries using weighted averages, with the result optionally scaled by amplitude before being written to the PWM hardware as a duty cycle, creating the illusion of an analog sine wave output through rapid digital switching.

- Table sizes are fixed and changing for frequency because at higher frequencies, a smaller table size is needed so phase increment does not select the same table entry.
    - Table index = Phase * Table size.