arm Education

Embedded Systems Essentials with Arm: Getting Started

Module 5

SV3 (5): Pulse Width Modulation

We will now introduce the concept of pulse-width modulation, but this topic will be covered in more detail in the next module. It is being used here for the purpose of the example.

Pulse width modulation or PWM is a simple method of using a rectangular digital waveform to create an analog output. PWM uses the width of the pulse to represent an amplitude.

The period of the wave is usually kept constant, and the pulse width or ON time is varied.

The duty cycle is the proportion of time that the pulse is ON or HIGH, and is expressed as a percentage.

Whatever duty cycle a PWM stream has, there is an average value. If the ON time is small, the average value is low; if the ON time is large, the average value is high. Therefore, by controlling the duty cycle, we control the average output value, represented by red line.

We can use the PwmOut Interface to control the frequency and duty cycle of a PWM signal. Here are some examples showing how to use the PWM.

The Mbed PwmOut Interface has a set of member functions which can be seen here. Note that all digital pins can be used as PWM outputs.

A sound is essentially an air wave. The amplitude and the frequency of the wave decide the volume and the pitch of the sound respectively. The speaker (or headphone) inputs electrical signals (voltages) and use them to turn a coil into an electromagnet, which can either attract or repel the magnet that moves back and forth. The motion of the magnet will further push and pull a diaphragm and create air waves (just like a drum).

In order to generate simple waveforms, we can use ‘for loops’. For example, for a sawtooth wave we could use the following code.

Or, for a triangle wave we can use two loops.

© Arm 2020