# 4 ANALOG OUTPUT – Code Snips

**Note: These Code Snips are taken straight from the book chapter; i.e. the “Program Examples”. In some cases therefore they are not complete programs.**

/\*Program Example 4.1: Three values of DAC are output in turn on Pin 18. Read the output on a DVM.

 \*/

#include "mbed.h"

AnalogOut Aout(p18); //create an analog output on pin 18

int main() {

 while(1) {

 Aout=0.25; // 0.25\*3.3V = 0.825V

 wait(2);

 Aout=0.5; // 0.5\*3.3V = 1.65V

 wait(2);

 Aout=0.75; // 0.75\*3.3V = 2.475V

 wait(2);

 }

}

****Program Example 4.1: Trial DAC output****

/\*Program Example 4.2: Saw tooth waveform on DAC output. View on oscilloscope

 \*/

#include "mbed.h"

AnalogOut Aout(p18);

float i;

int main() {

 while(1){

 for (i=0;i<1;i=i+0.1){ // i is incremented in steps of 0.1

 Aout=i;

 wait(0.001); // wait 1 millisecond

 }

 }

}

****Program Example 4.2: Saw tooth waveform****

/\*Program Example 4.3: Sine wave on DAC output. View on oscilloscope

 \*/

#include "mbed.h"

AnalogOut Aout(p18);

float i;

int main() {

 while(1) {

 for (i=0;i<2;i=i+0.05) {

 Aout=0.5+0.5\*sin(i\*3.14159); // Compute the sine value, + half the range

 wait(.001); // Controls the sine wave period

 }

 }

}

****Program Example 4.3: Generating a sinusoidal waveform****

/\*Sets PWM source to fixed frequency and duty cycle. Observe output on oscilloscope.

 \*/

#include "mbed.h"

PwmOut PWM1(p21); //create a PWM output called PWM1 on pin 21

int main() {

 PWM1.period(0.010); // set PWM period to 10 ms

 PWM1=0.5; // set duty cycle to 50%

}

****Program Example 4.4: Trial PWM output****

/\*Program Example 4.5: PWM control to DC motor is repeatedly ramped

 \*/

#include "mbed.h"

PwmOut PWM1(p21);

float i;

int main() {

 PWM1.period(0.010); //set PWM period to 10 ms

 while(1) {

 for (i=0;i<1;i=i+0.01) {

 PWM1=i; // update PWM duty cycle

 wait(0.2);

 }

 }

}

****Program Example 4.5: Controlling motor speed with mbed PWM source****

/\*Program Example 4.6: Software generated PWM. 2 PWM values generated in turn, with full on and off included for comparison.

 \*/

#include "mbed.h"

DigitalOut motor(p6);

int i;

int main() {

 while(1) {

 motor = 0; //motor switched off for 5 secs

 wait (5);

 for (i=0;i<5000;i=i+1) { //5000 PWM cycles, low duty cycle

 motor = 1;

 wait\_us(400); //output high for 400us

 motor = 0;

 wait\_us(600); //output low for 600us

 }

 for (i=0;i<5000;i=i+1) { //5000 PWM cycles, high duty cycle

 motor = 1;

 wait\_us(800); //output high for 800us

 motor = 0;

 wait\_us(200); //output low for 200us

 }

 motor = 1; //motor switched fully on for 5 secs

 wait (5);

 }

}

****Program Example 4.6: Generating PWM in software****

/\*Program Example 4.7: Plays the tune "Oranges and Lemons" on a piezo buzzer, using PWM

 \*/

#include "mbed.h"

PwmOut buzzer(p26);

 //frequency array

float frequency[]={659,554,659,554,440,494,554,587,494,659,554,440};

float beat[]={1,1,1,1,1,0.5,0.5,1,1,1,1,2}; //beat array

int main() {

  while (1) {

    for (int i=0;i<=11;i++) {

      buzzer.period(1/(2\*frequency[i]));    // set PWM period

      buzzer=0.5;                           // set duty cycle

      wait(0.4\*beat[i]);                    // hold for beat period

    }

  }

}

****Program Example 4.7: “Oranges and Lemons” program****