# 3 DIGITAL INPUT and 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 2.1: A program which flashes mbed LED1 on and off. Demonstrating use of digital output and wait functions. Taken from the mbed site. \*/

#include "mbed.h" //include the mbed header file as part of this program

// program variable myled is created, and linked with mbed LED1  
DigitalOut myled(LED1);

int main() { //the main function starts here

while(1) { //a continuous loop is created

myled = 1; //switch the led on, by setting the output to logic 1

wait(0.2); //wait 0.2 seconds

myled = 0; //switch the led off

wait(0.2); //wait 0.2 seconds

} //end of while loop

} //end of main function

**Program Example 2.1: Repeated (and commented) for convenience**

/\*Program Example 3.1: Demonstrates use of while loops. No external connection required

\*/

#include "mbed.h"

DigitalOut myled(LED1);

DigitalOut yourled(LED4);

int main() {

char i=0; //declare variable i, and set to 0

while(1){ //start endless loop

while(i<10) { //start first conditional while loop

myled = 1;

wait(0.2);

myled = 0;

wait(0.2);

i = i+1; //increment i

} //end of first conditional while loop

while(i>0) { //start second conditional loop

yourled = 1;

wait(0.2);

yourled = 0;

wait(0.2);

i = i-1;

}

} //end infinite loop block

} //end of main

Program Example 3.1: Using *while*

/\*Program Example 3.2: Flashes red and green LEDs in simple time-based pattern

\*/

#include "mbed.h"

DigitalOut redled(p5); //define and name a digital output on pin 5

DigitalOut greenled(p6); //define and name a digital output on pin 6

int main() {

while(1) {

redled = 1;

greenled = 0;

wait(0.2);

redled = 0;

greenled = 1;

wait(0.2);

}

}

****Program Example 3.2: Flashing external LEDs****

/\*Program Example 3.3: Flashes one of two LEDs, depending on the state of a 2-way switch

\*/

#include "mbed.h"

DigitalOut redled(p5);

DigitalOut greenled(p6);

DigitalIn switchinput(p7);

int main() {

while(1) {

if (switchinput==1) { //test value of switchinput

//execute following block if switchinput is 1

greenled = 0; //green led is off

redled = 1; // flash red led

wait(0.2);

redled = 0;

wait(0.2);

} //end of if

else { //here if switchinput is 0

redled = 0; //red led is off

greenled = 1; // flash green led

wait(0.2);

greenled = 0;

wait(0.2);

} //end of else

} //end of while(1)

} //end of main

****Program Example 3.3: Using *if* and elseto respond to external switch****

/\*Program Example 3.4.

Uses Joystick values to switch tri-colour led \*/

#include "mbed.h"

DigitalOut redled(p23); //leds in tri-colour led

DigitalOut greenled(p24);

DigitalOut blueled(p25);

DigitalIn joyleft(p13); //Joystick left

DigitalIn joyright(p16); //Joystick right

int main() {

greenled=redled=blueled=1; //switch all leds off (logic 1 for off)

redled = 0; //switch red led on, diagnostic (logic 0 for on)

wait(1);

redled = 1; //switch red led off

while(1) {

if (joyleft==1) { //test if the joystick is pushed left

greenled = 0; //switch green led on

wait(1); //wait one second

greenled = 1; //switch green led off

}

if (joyright==1) { //test if the joystick is pushed right

redled = 0; //switch red led on

wait(1);

redled = 1; //switch red led off

}

}

}

**Program Example 3.4: Controlling application board LED with the joystick**

/\*Program Example 3.5: Transfers the value of the joystick to mbed LEDs

\*/

#include "mbed.h"

BusIn joystick(p15,p12,p13,p16);

DigitalIn fire(p14);

BusOut leds(LED1,LED2,LED3,LED4);

int main(){

    while(1){

        if (fire) {

            leds=0xf;

        }

else {

            leds=joystick;

        }

        wait(0.1);

    }

}

****Program Example 3.5: Controlling mbed LEDs from the app board Joystick (reproduced from**** Reference 2.6)

/\*Program Example 3.6: Simple program to test KTIR slotted opto-sensor. Switches an LED according to state of sensor

\*/

#include "mbed.h"

DigitalOut redled(p5);

DigitalIn opto\_switch(p12);

int main() {

while(1) {

if (opto\_switch==1) //input = 1 if beam interrupted

redled = 1; //switch led on if beam interrupted

else

redled = 0; //led off if no interruption

} //end of while

}

****Program Example 3.6: Applying the photo-interrupter****

/\*Program Example 3.7: Simple demonstration of 7-segment display. Displays digits 0, 1, 2, 3 in turn.

\*/

#include "mbed.h"

BusOut display(p5,p6,p7,p8,p9,p10,p11,p12); // segments dp,a,b,c,d,e,f,g

int main() {

while(1) {

for(int i=0; i<4; i++) {

switch (i){

case 0: display = 0x3F; break; //display 0

case 1: display = 0x06; break; //display 1

case 2: display = 0x5B; break;

case 3: display = 0x4F; break;

} //end of switch

wait(0.2);

} //end of for

} //end of while

} //end of main

****Program Example 3.7: Using *for* to sequence values to a seven-segment display****