# 6 FURTHER PROGRAMMING TECHNIQUES – Code Snips

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

/\* Program Example 6.1: seven-segment display counter

\*/

#include "mbed.h"

BusOut Seg1(p5,p6,p7,p8,p9,p10,p11,p12); // A,B,C,D,E,F,G,DP

char SegConvert(char SegValue); // function prototype

char A=0; // declare variables A and B

char B;

int main() { // main program

while (1) { // infinite loop

B=SegConvert(A); // Call function to return B

Seg1=B; // Output B

A++; // increment A

if (A>0x09){ // if A > 9 reset to zero

A=0;

}

wait(0.5); // delay 500 milliseconds

}

}

char SegConvert(char SegValue) { // function 'SegConvert'

char SegByte=0x00;

switch (SegValue) { //DP G F E D C B A

case 0 : SegByte = 0x3F;break; // 0 0 1 1 1 1 1 1 binary

case 1 : SegByte = 0x06;break; // 0 0 0 0 0 1 1 0 binary

case 2 : SegByte = 0x5B;break; // 0 1 0 1 1 0 1 1 binary

case 3 : SegByte = 0x4F;break; // 0 1 0 0 1 1 1 1 binary

case 4 : SegByte = 0x66;break; // 0 1 1 0 0 1 1 0 binary

case 5 : SegByte = 0x6D;break; // 0 1 1 0 1 1 0 1 binary

case 6 : SegByte = 0x7D;break; // 0 1 1 1 1 1 0 1 binary

case 7 : SegByte = 0x07;break; // 0 0 0 0 0 1 1 1 binary

case 8 : SegByte = 0x7F;break; // 0 1 1 1 1 1 1 1 binary

case 9 : SegByte = 0x6F;break; // 0 1 1 0 1 1 1 1 binary

}

return SegByte;

}

Program Example 6.1: Seven-segment display counter

/\* Program Example 6.2: Display counter for 0-99

\*/

#include "mbed.h"

BusOut Seg1(p5,p6,p7,p8,p9,p10,p11,p12); // A,B,C,D,E,F,G,DP

BusOut Seg2(p13,p14,p15,p16,p17,p18,p19,p20);

char SegConvert(char SegValue); // function prototype

int main() { // main program

while (1) { // infinite loop

for (char j=0;j<10;j++) { // counter loop 1

Seg2=SegConvert(j); // tens column

for (char i=0;i<10;i++) { // counter loop 2

Seg1=SegConvert(i); // units column

wait(0.2);

}

}

}

}

// add SegConvert function here...

Program Example 6.2: Two digit seven-segment display counter.

/\* Program Example 6.3: Host keypress to 7-seg display

\*/

#include "mbed.h"

Serial pc(USBTX, USBRX); // comms to host PC

BusOut Seg1(p5,p6,p7,p8,p9,p10,p11,p12); // A,B,C,D,E,F,G,DP

BusOut Seg2(p13,p14,p15,p16,p17,p18,p19,p20); // A,B,C,D,E,F,G,DP

void SegInit(void); // function prototype

void HostInit(void); // function prototype

char GetKeyInput(void); // function prototype

char SegConvert(char SegValue); // function prototype

char data1, data2; // variable declarations

int main() { // main program

SegInit(); // call function to initialise the 7-seg displays

HostInit(); // call function to initialise the host terminal

while (1) { // infinite loop

data2 = GetKeyInput(); // call function to get 1st key press

Seg2=SegConvert(data2); // call function to convert and output

data1 = GetKeyInput(); // call function to get 2nd key press

Seg1=SegConvert(data1); // call function to convert and output

pc.printf(" "); // display spaces between numbers

}

}

// functions

void SegInit(void) {

Seg1=SegConvert(0); // initialise to zero

Seg2=SegConvert(0); // initialise to zero

}

void HostInit(void) {

pc.printf("\n\rType two digit numbers to be displayed\n\r");

}

char GetKeyInput(void) {

char c = pc.getc(); // get keyboard data (ascii 0x30-0x39)

pc.printf("%c",c); // print ascii value to host PC terminal

return (c&0x0F); // apply bit mask to convert to decimal, and return

}

// copy SegConvert function here too...

Program Example 6.3: Two digit seven-segment display based on host key presses

/\* Program Example 6.4: Template for .h header file

\*/

#ifndef VARIABLE\_H // if VARIABLE\_H has not previously been defined

#define VARIABLE\_H // define it now

// header declarations here…

#endif // end of the if directive

Program Example 6.4: Example header file template

/\* Program Example 6.5: main.cpp file for modular 7-seg keyboard controller

\*/

#include "mbed.h"

#include "HostIO.h"

#include "SegDisplay.h"

char data1, data2; // variable declarations

int main() { // main program

SegInit(); // call init function

HostInit(); // call init function

while (1) { // infinite loop

data2 = GetKeyInput(); // call to get 1st key press

Seg2 = SegConvert(data2); // call to convert and output

data1 = GetKeyInput(); // call to get 2nd key press

Seg1 = SegConvert(data1); // call to convert and output

pc.printf(" "); // display spaces on host

}

}

Program Example 6.5: Source code for main.cpp

/\* Program Example 6.6: SegDisplay.cpp file for modular 7-seg keyboard controller

\*/

#include "SegDisplay.h"

BusOut Seg1(p5,p6,p7,p8,p9,p10,p11,p12); // A,B,C,D,E,F,G,DP

BusOut Seg2(p13,p14,p15,p16,p17,p18,p19,p20); // A,B,C,D,E,F,G,DP

void SegInit(void) {

Seg1=SegConvert(0); // initialise to zero

Seg2=SegConvert(0); // initialise to zero

}

char SegConvert(char SegValue) { // function 'SegConvert'

char SegByte=0x00;

switch (SegValue) { //DP G F E D C B A

case 0 : SegByte = 0x3F; break; // 0 0 1 1 1 1 1 1 binary

case 1 : SegByte = 0x06; break; // 0 0 0 0 0 1 1 0 binary

case 2 : SegByte = 0x5B; break; // 0 1 0 1 1 0 1 1 binary

case 3 : SegByte = 0x4F; break; // 0 1 0 0 1 1 1 1 binary

case 4 : SegByte = 0x66; break; // 0 1 1 0 0 1 1 0 binary

case 5 : SegByte = 0x6D; break; // 0 1 1 0 1 1 0 1 binary

case 6 : SegByte = 0x7D; break; // 0 1 1 1 1 1 0 1 binary

case 7 : SegByte = 0x07; break; // 0 0 0 0 0 1 1 1 binary

case 8 : SegByte = 0x7F; break; // 0 1 1 1 1 1 1 1 binary

case 9 : SegByte = 0x6F; break; // 0 1 1 0 1 1 1 1 binary

}

return SegByte;

}

Program Example 6.6: Source code for SegDisplay.cpp

/\* Program Example 6.7: SegDisplay.h file for modular 7-seg keyboard controller

\*/

#ifndef SEGDISPLAY\_H

#define SEGDISPLAY\_H

#include "mbed.h"

extern BusOut Seg1; // allow Seg1 to be manipulated by other files

extern BusOut Seg2; // allow Seg2 to be manipulated by other files

void SegInit(void); // function prototype

char SegConvert(char SegValue); // function prototype

#endif

Program Example 6.7: Source code for SegDisplay.cpp

/\* Program Example 6.8: HostIO.cpp code for modular 7-seg keyboard controller

\*/

#include "HostIO.h"

Serial pc(USBTX, USBRX); // communication to host PC

void HostInit(void) {

pc.printf("\n\rType two digit numbers to be \n\r");

}

char GetKeyInput(void) {

char c = pc.getc(); // get keyboard ascii data

pc.printf("%c",c); // print ascii value to host PC terminal

return (c&0x0F); // return value as non-ascii

}

Program Example 6.8: Source code for HostIO.cpp

/\* Program Example 6.9: HostIO.h code for modular 7-seg keyboard controller

\*/

#ifndef HOSTIO\_H

#define HOSTIO\_H

#include "mbed.h"

extern Serial pc; // allow pc to be manipulated by other files

void HostInit(void); // function prototype

char GetKeyInput(void); // function prototype

#endif

Program Example 6.9: Source code for HostIO.h