/\* ###################################################################

\*\* Filename : main.c

\*\* Project : ProjectExam

\*\* Processor : MK20DX128VLH5

\*\* Version : Driver 01.01

\*\* Compiler : GNU C Compiler

\*\* Date/Time : 2019-11-10, 17:00, # CodeGen: 0

\*\* Abstract :

\*\* Main module.

\*\* This module contains user's application code.

\*\* Settings :

\*\* Contents :

\*\* No public methods

\*\*

\*\* ###################################################################\*/

/\*!

\*\* @file main.c

\*\* @version 01.01

\*\* @brief

\*\* Main module.

\*\* This module contains user's application code.

\*/

/\*!

\*\* @addtogroup main\_module main module documentation

\*\* @{

\*/

/\* MODULE main \*/

/\* Including needed modules to compile this module/procedure \*/

**#include** "Cpu.h"

**#include** "Events.h"

**#include** "Term1.h"

**#include** "Inhr1.h"

**#include** "ASerialLdd1.h"

**#include** "AD1.h"

**#include** "AdcLdd1.h"

**#include** "Bit1.h"

**#include** "BitIoLdd1.h"

**#include** "FC321.h"

**#include** "RealTimeLdd1.h"

**#include** "TU1.h"

/\* Including shared modules, which are used for whole project \*/

**#include** "PE\_Types.h"

**#include** "PE\_Error.h"

**#include** "PE\_Const.h"

**#include** "IO\_Map.h"

/\* User includes (#include below this line is not maintained by Processor Expert) \*/

**#include** <stdlib.h>

/\*lint -save -e970 Disable MISRA rule (6.3) checking. \*/

**int** **main**(**void**)

/\*lint -restore Enable MISRA rule (6.3) checking. \*/

{

/\* Write your local variable definition here \*/

/\*\*\* Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! \*\*\*/

PE\_low\_level\_init();

/\*\*\* End of Processor Expert internal initialization. \*\*\*/

/\* Write your code here \*/

/\* For example: for(;;) { } \*/

**void** **send\_string**(**const** **char** \*str){

**int** len, i;

len = **strlen**(str);

**for**(i = 0; i < len; i++){

Term1\_SendChar(str[i]);

}

}

**int** quit;

**int** score;

**char** c;

**int** answer;

uint16 xmeasure;

**float** xsmooth;

**int** time;

**int** a;

**int** b;

**int** min;

**int** value;

**int** easy;

**int** n;

score = 0;

easy = 1;

xsmooth = 0;

send\_string("CC2511 Exam 2018\r\nAlex Philp");

AD1\_Calibrate(TRUE);

AD1\_Measure(TRUE);

AD1\_GetValue16(&xmeasure);

**srand**(xmeasure);

**for**(;;){

Term1\_ReadChar(&c);

**if**(c == 'm'){

send\_string("Monitoring Mode\r\n");

quit = 0;

**do**{

AD1\_Measure(TRUE);

AD1\_GetValue16(&xmeasure);

xsmooth = 0.9\*xsmooth + 0.1\*xmeasure;

Term1\_SendFloatNum(xsmooth);

send\_string("\r\n");

Term1\_SendFloatNum(xmeasure);

send\_string("\r\n");

**if**(xsmooth > xmeasure){

Bit1\_SetVal();

}

**else**{

Bit1\_ClrVal();

}

FC321\_Reset();

FC321\_GetTimeMS(&time);

**while**(time < 200){

FC321\_GetTimeMS(&time);

Term1\_ReadChar(&c);

**if**(Term1\_KeyPressed()){

quit = 1;

**break**;

}

}

}**while**(quit == 0);

}

**else** **if**(c == 'g'){

send\_string("Entering game mode\r\n");

**while**(score < 10){

send\_string("Score:");

Term1\_SendNum(score);

send\_string("\r\n");

/\*if(score >= 0){

PWM2\_SetRatio16(51\*score);

PWM3\_SetRatio16(0);

}

else if(score < 0){

PWM3\_SetRatio16(-51\*score);

PWM2\_SetRatio16(0);

}\*/

**if**(easy = 1){

a = **rand**() % 1000;

min = a - 9;

value = **rand**() % 9;

b = min + value;

Term1\_SendNum(a);

send\_string("-");

Term1\_SendNum(b);

FC321\_Reset();

**while**(answer == 0){

//timer

FC321\_GetTimeMS(&time);

**if**(time >10000){

answer = 1;

score = score - 1;

send\_string("Time out\r\n");

**break**;

}

//key press

**if**(Term1\_KeyPressed()){

answer = 1;

Term1\_ReadChar(&c); //check typecasting rules

**if**((int)c == a-b){

score++;

send\_string("Correct\r\n");

**if**(score = 5){

easy = 0;

}

}

**else**{

score = score - 1;

send\_string("Incorrect\r\n");

}

}

}

}

**else**{

a = **rand**() % 1000;

n = **rand**() % 1000;

value = **rand**() % 9;

b = ((a+n)-9) + value;

Term1\_SendNum(a);

send\_string("-");

Term1\_SendNum(b);

FC321\_Reset();

**while**(answer = 0){

//timer

FC321\_GetTimeMS(&time);

**if**(time >10000){

answer = 1;

score = score - 1;

send\_string("Time out\r\n");

**break**;

}

//key press

**if**(Term1\_KeyPressed()){

answer = 1;

Term1\_ReadChar(&c); //check typecasting rules

**if**((int)c == a-b){

score++;

send\_string("Correct\r\n");

**if**(score = 5){

easy = 0;

}

}

**else**{

score = score - 1;

send\_string("Incorrect\r\n");

}

}

}

}

}

}

}

/\*\*\* Don't write any code pass this line, or it will be deleted during code generation. \*\*\*/

/\*\*\* RTOS startup code. Macro PEX\_RTOS\_START is defined by the RTOS component. DON'T MODIFY THIS CODE!!! \*\*\*/

**#ifdef** PEX\_RTOS\_START

PEX\_RTOS\_START(); /\* Startup of the selected RTOS. Macro is defined by the RTOS component. \*/

**#endif**

/\*\*\* End of RTOS startup code. \*\*\*/

/\*\*\* Processor Expert end of main routine. DON'T MODIFY THIS CODE!!! \*\*\*/

**for**(;;){}

/\*\*\* Processor Expert end of main routine. DON'T WRITE CODE BELOW!!! \*\*\*/

} /\*\*\* End of main routine. DO NOT MODIFY THIS TEXT!!! \*\*\*/

/\* END main \*/

/\*!

\*\* @}

\*/

/\*

\*\* ###################################################################

\*\*

\*\* This file was created by Processor Expert 10.5 [05.21]

\*\* for the Freescale Kinetis series of microcontrollers.

\*\*

\*\* ###################################################################

\*/