if ((int) key <= 57){

xil\_printf("Key Pressed: %d\r\n", ((int) key-48)\*((int)key-48));

}

else{

xil\_printf("Key Pressed: %d\r\n", ((int) key-55)\*((int)key-55));

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* \*/

/\* PmodKYPD.c -- Demo for the use of the Pmod Keypad IP core \*/

/\* \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Author: Mikel Skreen \*/

/\* Copyright 2016, Digilent Inc. \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* File Description: \*/

/\* \*/

/\* This demo continuously captures keypad data and prints a message to an \*/

/\* attached serial terminal whenever a positive edge is detected on any of \*/

/\* the sixteen keys. In order to receive messages, a serial terminal \*/

/\* application on your PC should be connected to the appropriate COM port for \*/

/\* the micro-USB cable connection to your board's USBUART port. The terminal \*/

/\* should be configured with 8-bit data, no parity bit, 1 stop bit, and the \*/

/\* the appropriate Baud rate for your application. If you are using a Zynq \*/

/\* board, use a baud rate of 115200, if you are using a MicroBlaze system, \*/

/\* use the Baud rate specified in the AXI UARTLITE IP, typically 115200 or \*/

/\* 9600 Baud. \*/

/\* \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Revision History: \*/

/\* \*/

/\* 06/08/2016(MikelS): Created \*/

/\* 08/17/2017(artvvb): Validated for Vivado 2015.4 \*/

/\* 08/30/2017(artvvb): Validated for Vivado 2016.4 \*/

/\* Added Multiple keypress error detection \*/

/\* 01/27/2018(atangzwj): Validated for Vivado 2017.4 \*/

/\* \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "PmodKYPD.h"

#include "sleep.h"

#include "xil\_cache.h"

#include "xparameters.h"

void DemoInitialize();

void DemoRun();

void DemoCleanup();

void DisableCaches();

void EnableCaches();

void DemoSleep(u32 millis);

PmodKYPD myDevice;

int main(void) {

DemoInitialize();

DemoRun();

DemoCleanup();

return 0;

}

// keytable is determined as follows (indices shown in Keypad position below)

// 12 13 14 15

// 8 9 10 11

// 4 5 6 7

// 0 1 2 3

#define DEFAULT\_KEYTABLE "0FED789C456B123A"

void DemoInitialize() {

EnableCaches();

KYPD\_begin(&myDevice, XPAR\_PMODKYPD\_0\_AXI\_LITE\_GPIO\_BASEADDR);

KYPD\_loadKeyTable(&myDevice, (u8\*) DEFAULT\_KEYTABLE);

}

void DemoRun() {

u16 keystate;

XStatus status, last\_status = KYPD\_NO\_KEY;

u8 key, last\_key = 'x';

// Initial value of last\_key cannot be contained in loaded KEYTABLE string

Xil\_Out32(myDevice.GPIO\_addr, 0xF);

xil\_printf("Pmod KYPD demo started. Press any key on the Keypad.\r\n");

while (1) {

// Capture state of each key

keystate = KYPD\_getKeyStates(&myDevice);

// Determine which single key is pressed, if any

status = KYPD\_getKeyPressed(&myDevice, keystate, &key);

// Print key detect if a new key is pressed or if status has changed

if (status == KYPD\_SINGLE\_KEY

&& (status != last\_status || key != last\_key)) {

if ((int) key <= 57){

xil\_printf("Key Pressed: %d\r\n", ((int) key-48)\*((int)key-48));

}

else{

xil\_printf("Key Pressed: %d\r\n", ((int) key-55)\*((int)key-55));

}

last\_key = key;

} else if (status == KYPD\_MULTI\_KEY && status != last\_status)

xil\_printf("Error: Multiple keys pressed\r\n");

last\_status = status;

usleep(1000);

}

}

void DemoCleanup() {

DisableCaches();

}

void EnableCaches() {

#ifdef \_\_MICROBLAZE\_\_

#ifdef XPAR\_MICROBLAZE\_USE\_ICACHE

Xil\_ICacheEnable();

#endif

#ifdef XPAR\_MICROBLAZE\_USE\_DCACHE

Xil\_DCacheEnable();

#endif

#endif

}

void DisableCaches() {

#ifdef \_\_MICROBLAZE\_\_

#ifdef XPAR\_MICROBLAZE\_USE\_DCACHE

Xil\_DCacheDisable();

#endif

#ifdef XPAR\_MICROBLAZE\_USE\_ICACHE

Xil\_ICacheDisable();

#endif

#endif

}