```
/*
 * Author: John Monahan S100Computers.com
 * Test for Edison TXB0108 Adaptor board.
 */

#include "mraa.h"
#include <stdio.h>
#include <unistd.h>
#include <string.h>

#define FALSE 0
#define TRUE 1
#define LOW 0
#define HIGH 1
#define ESC  0x1b
#define CR   0x0d
#define LF   0x0a
#define BS   0x08
#define BELL 0x07
#define SP   0x20
#define DEL  0x7f

#define IOBYTE  0xEF                    // S100 Bus IOBYTE Port on V3 SMB Board
#define CON_OUT_PORT 1                  // Port 1 on Propeller driven Console I/O Board
#define CON_IN_PORT 1                   // Port 1 on Propeller driven Console I/O Board
#define CON_STATUS_PORT 0               // Port 0 on Propeller driven Console I/O Board

#define E_STOP_REQUEST 4                // GP135, Input low to high stops S100 process on Edison
#define E_sINP 13                       // GP128, For S100 bus sINP
#define E_sOUT 14                       // GP13_PWM1
#define E_MEMR 15                       // GP165
#define E_MEMW 19                       // GP19
#define P14 20                          // GP12_PWM0  (Unused, seems inactive)
#define DATA_WR 21                      // GP183_PWM3
#define EDISON_READY 23                 // GP110
#define DATA_RD 24                      // GP114

#define ACTIVATE_EDISON 25              // GP129
#define RW_PULSE 38                     // GP43
#define P15  44                         // GP134       (Unused, seems inactive)
#define USB_PC6 55                      // GP81        USB Out Status port bit (Read only)
#define P16  39                         // GP77        (Unused, seems inactive)
#define USB_PC7 40                      // GP81        USB In Status port bit (Read only)
#define E_PSYNC 41                      // GP83
```

```
#define USB_SEL 45                          // GP45         (USB Data & Status port select)
#define bDO1 46                             // GP47         (All set as OUTPUTS)
#define bDO2 47                             // GP49
#define bDO3 48                             // GP15
#define bDO4 49                             // GP48
#define bDO5 50                             // GP42
#define bDO6 51                             // GP42
#define bDO7 52                             // GP78

#define E_WAIT 53                           // GP79          Either adds wait states to S100 bus or resets
back to Z80
#define ADDRESS3  54                        // GP80
#define ADDRESS2  31                        // GP44
#define ADDRESS1  32                        // GP46
#define E_sINTA   33                        // GP48
#define E_TO_BUS  35                        // GP131         (Note, Edison Output only pin)
#define E_S100_INT 36                       // GP14
#define bDO0 37                             // GP40          (Set as OUTPUT)

#define bDI0 0                              // GP182_PWM2    (All set as INPUTS)
#define bDI1 26                             // GP130
#define bDI2 6                              // GP27
#define bDI3 7                              // GP20
#define bDI4 8                              // GP28
#define bDI5 9                              // GP111
#define bDI6 10                             // GP109
#define bDI7 11                             // GP115

mraa_gpio_context pin[56];

int main()
{
int  i;
mraa_init();

printf("Start Edison Adaptor board test \n");

      for (i=0; i < 56; i++)                // INITILIZE ALL EDISON PINS
            {
            switch(i)
                  {
                  case 1:                   //Skip these pins (Note these are MRAA library pin numbers)
                  case 2:
                  case 3:
                  case 5:                   // GP27 (Unused, seems inactive)
```

```
                case 12:                                    // GP12_PWM0 (P14)
                case 16:
                case 17:
                case 18:
                case 22:
                case 27:
                case 28:
                case 29:
                case 30:
                case 34:
                case 42:
                case 43:
                      break;

                default:
                      pin[i] = mraa_gpio_init(i);                   // Default all outputs, initially HIGH
                      mraa_gpio_mode(pin[i],MRAA_GPIO_STRONG);      // Note includes 8 Data outputs (U4) and address
lines
                      mraa_gpio_dir(pin[i], MRAA_GPIO_OUT_HIGH);
                      mraa_gpio_use_mmaped(pin[i],1);               // For fast I/O
                      }
            }

//    while(TRUE)                                                   // <--- DIGNOSTIC TEST (Loops forever if active)
//          {
//          mraa_gpio_write(pin[bDO0],HIGH);                        //Pin 16, (37, GP40)
//          usleep(10000);
//          mraa_gpio_write(pin[bDO0],LOW);                         //Pin 16, (37, GP40)
//          usleep(10000);
//          }

      while(TRUE)                                                   // <--- DIGNOSTIC TEST (Loops forever if active)
            {
            mraa_gpio_write(pin[ACTIVATE_EDISON],HIGH);             //Pin 1 (25, GP129)
            mraa_gpio_write(pin[E_STOP_REQUEST],HIGH);              //Pin 3 (5, GP135)
            mraa_gpio_write(pin[RW_PULSE],HIGH);                    //Pin 5 (38, GP43)
            mraa_gpio_write(pin[P15],HIGH);                         //Pin 7, (44, GP134) Pin is inactive, always low!
            mraa_gpio_write(pin[USB_PC6],HIGH);                     //Pin 9, (55,GP81)
            mraa_gpio_write(pin[P16],HIGH);                         //Pin 11, (39, GP77) Pin is inactive, always low!
            mraa_gpio_write(pin[USB_PC7],HIGH);                     //Pin 13, (40, GP82)
            mraa_gpio_write(pin[E_PSYNC],HIGH);                     //Pin 15, (41, GP83)
            mraa_gpio_write(pin[E_sINP],HIGH);                      //Pin 17, (13, GP128)
            mraa_gpio_write(pin[E_sOUT],HIGH);                      //Pin 19, (14, GP13_PWM1)
            mraa_gpio_write(pin[E_MEMR],HIGH);                      //Pin 21, (15, GP165)
            mraa_gpio_write(pin[E_MEMW],HIGH);                      //Pin 23, (19, GP19)
```

```
mraa_gpio_write(pin[P14],HIGH);                 //Pin 25, (20, GP12_PWM0) Pin currently functional
mraa_gpio_write(pin[DATA_WR],HIGH);             //Pin 27, (21, GP183_PWM3)
mraa_gpio_write(pin[EDISON_READY],HIGH);        //Pin 29, (23, GP110)
mraa_gpio_write(pin[DATA_RD],HIGH);             //Pin 31, (24, GP114)

mraa_gpio_write(pin[bDI0],HIGH);                //Pin 33, (0, GP182_PWM2)
mraa_gpio_write(pin[bDI1],HIGH);                //Pin 35, (26, GP130)
mraa_gpio_write(pin[bDI2],HIGH);                //Pin 37, (6, GP27)
mraa_gpio_write(pin[bDI3],HIGH);                //Pin 39, (8, GP20)
mraa_gpio_write(pin[bDI4],HIGH);                //Pin 41, (9, GP28)
mraa_gpio_write(pin[bDI5],HIGH);                //Pin 43, (10, GP111)
mraa_gpio_write(pin[bDI6],HIGH);                //Pin 45, (11, GP109)
mraa_gpio_write(pin[bDI7],HIGH);                //Pin 47, (12, GP115)

mraa_gpio_write(pin[E_WAIT],HIGH);              //Pin 2   (53, GP79)
mraa_gpio_write(pin[ADDRESS3],HIGH);            //Pin 4   (54, GP80)
mraa_gpio_write(pin[ADDRESS2],HIGH);            //Pin 6   (31, GP44)
mraa_gpio_write(pin[ADDRESS1],HIGH);            //Pin 8,  (32, GP46)
mraa_gpio_write(pin[E_sINTA],HIGH);             //Pin 10, (33,GP48)
mraa_gpio_write(pin[E_TO_BUS],HIGH);            //Pin 12, (35, GP131)
mraa_gpio_write(pin[E_S100_INT],HIGH);          //Pin 14, (36, GP14)
mraa_gpio_write(pin[bDO0],HIGH);                //Pin 16, (37, GP40)
mraa_gpio_write(pin[USB_SEL],HIGH);             //Pin 18, (45, GP45)

mraa_gpio_write(pin[bDO1],HIGH);                //Pin 20, (46, GP47)
mraa_gpio_write(pin[bDO2],HIGH);                //Pin 22, (47, GP49)
mraa_gpio_write(pin[bDO3],HIGH);                //Pin 24, (48, GP15)
mraa_gpio_write(pin[bDO4],HIGH);                //Pin 26, (49, GP84)
mraa_gpio_write(pin[bDO5],HIGH);                //Pin 28, (50, GP42)
mraa_gpio_write(pin[bDO6],HIGH);                //Pin 30, (51, GP41)
mraa_gpio_write(pin[bDO7],HIGH);                //Pin 32, (52, GP78)

usleep(10000);

mraa_gpio_write(pin[ACTIVATE_EDISON],LOW);      //Pin 1 (25, GP129)
mraa_gpio_write(pin[E_STOP_REQUEST],LOW);       //Pin 3 (5, GP135)
mraa_gpio_write(pin[RW_PULSE],LOW);             //Pin 5 (38, GP43)
mraa_gpio_write(pin[P15],LOW);                  //Pin 7, (44, GP134) Pin is inactive, always low!
mraa_gpio_write(pin[USB_PC6],LOW);              //Pin 9, (55,GP81)
mraa_gpio_write(pin[P16],LOW);                  //Pin 11, (39, GP77) Pin is inactive, always low!
mraa_gpio_write(pin[USB_PC7],LOW);              //Pin 13, (40, GP82)
mraa_gpio_write(pin[E_PSYNC],LOW);              //Pin 15, (41, GP83)
mraa_gpio_write(pin[E_sINP],LOW);               //Pin 17, (13, GP128)
mraa_gpio_write(pin[E_sOUT],LOW);               //Pin 19, (14, GP13_PWM1)
mraa_gpio_write(pin[E_MEMR],LOW);               //Pin 21, (15, GP165)
```

```
        mraa_gpio_write(pin[E_MEMW],LOW);                   //Pin 23, (19, GP19)
        mraa_gpio_write(pin[P14],LOW);                      //Pin 25, (20, GP12_PWM0) Pin currently functional
        mraa_gpio_write(pin[DATA_WR],LOW);                  //Pin 27, (21, GP183_PWM3)
        mraa_gpio_write(pin[EDISON_READY],LOW);             //Pin 29, (23, GP110)
        mraa_gpio_write(pin[DATA_RD],LOW);                  //Pin 31, (24, GP114)

        mraa_gpio_write(pin[bDI0],LOW);                     //Pin 33, (0, GP182_PWM2)
        mraa_gpio_write(pin[bDI1],LOW);                     //Pin 35, (26, GP130)
        mraa_gpio_write(pin[bDI2],LOW);                     //Pin 37, (6, GP27)
        mraa_gpio_write(pin[bDI3],LOW);                     //Pin 39, (8, GP20)
        mraa_gpio_write(pin[bDI4],LOW);                     //Pin 41, (9, GP28)
        mraa_gpio_write(pin[bDI5],LOW);                     //Pin 43, (10, GP111)
        mraa_gpio_write(pin[bDI6],LOW);                     //Pin 45, (11, GP109)
        mraa_gpio_write(pin[bDI7],LOW);                     //Pin 47, (12, GP115)

        mraa_gpio_write(pin[E_WAIT],LOW);                   //Pin 2   (53, GP79)
        mraa_gpio_write(pin[ADDRESS3],LOW);                 //Pin 4   (54, GP80)
        mraa_gpio_write(pin[ADDRESS2],LOW);                 //Pin 6   (31, GP44)
        mraa_gpio_write(pin[ADDRESS1],LOW);                 //Pin 8,  (32, GP46)
        mraa_gpio_write(pin[E_sINTA],LOW);                  //Pin 10, (33,GP48)
        mraa_gpio_write(pin[E_TO_BUS],LOW);                 //Pin 12, (35, GP131)
        mraa_gpio_write(pin[E_S100_INT],LOW);               //Pin 14, (36, GP14)
        mraa_gpio_write(pin[bDO0],LOW);                     //Pin 16, (37, GP40)
        mraa_gpio_write(pin[USB_SEL],LOW);                  //Pin 18, (45, GP45)

        mraa_gpio_write(pin[bDO1],LOW);                     //Pin 20, (46, GP47)
        mraa_gpio_write(pin[bDO2],LOW);                     //Pin 22, (47, GP49)
        mraa_gpio_write(pin[bDO3],LOW);                     //Pin 24, (48, GP15)
        mraa_gpio_write(pin[bDO4],LOW);                     //Pin 26, (49, GP84)
        mraa_gpio_write(pin[bDO5],LOW);                     //Pin 28, (50, GP42)
        mraa_gpio_write(pin[bDO6],LOW);                     //Pin 30, (51, GP41)
        mraa_gpio_write(pin[bDO7],LOW);                     //Pin 32, (52, GP78)

        usleep(10000);
        printf("S100_Edison Flash Test Running.\n");
        }
    return MRAA_SUCCESS;
}
```