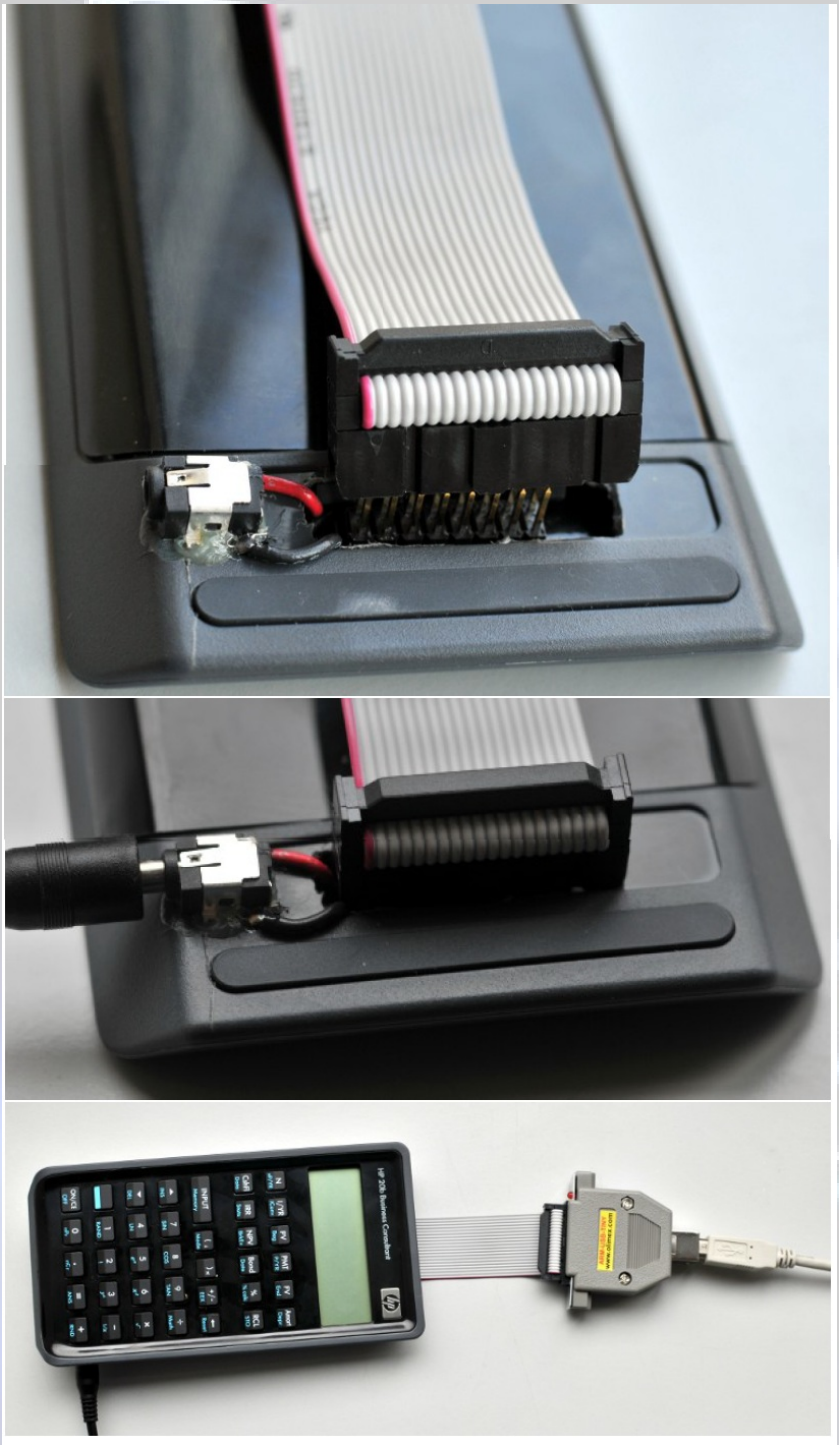
ENGI E1112 Department Project Report: Computer Science/Computer Engineering

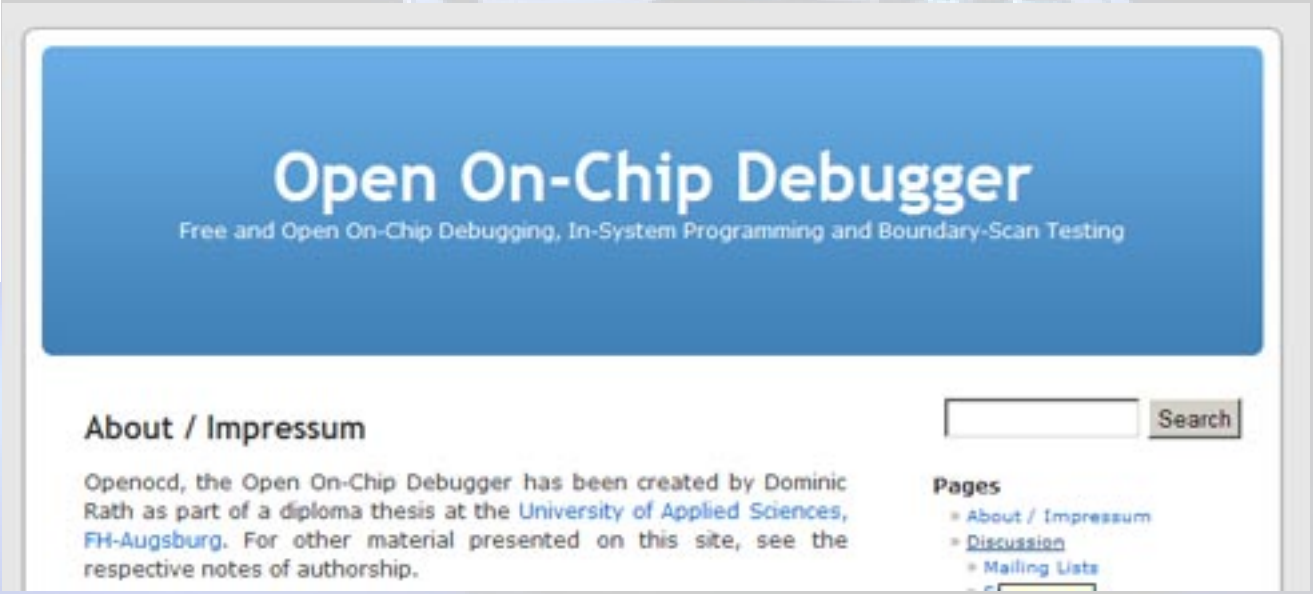
Ankita Gore, Christina Huang, Shikhar Kumar
December 16, 2011

Platform



Software Architecture

ubuntu



Tutorial

Stack Level 4		-15
Stack Level 3	-15	12
Stack Level 2	12	41
Stack Level 1	41	23

-15
-15
12
64

Software Details - Lab 1

```
1  //Ankita Gore, Shikhar Kumar, Christina Huang
2  #include "AT91SAM7L128.h"
3  #include "lcd.h"
4
5  #define DELAY 50000          //Delay factor
6  #define STRLEN 19            //Length of display string
7  #define COLS 15              //Number of spaces allotted to display on the LCD
8  #define BLANKSPACES 2        //Number of blank spaces to display before wrapping around
9
10 int main()
11 {
12   *AT91C_WDTC_WDMR = AT91C_WDTC_WDDIS; //Turn off watchdog timer
13   lcd_init();
14
15   char name1[] = "HELLO WORLD ABC123";      //Display string
16   char name2[COLS];                         //String which will be displayed on LCD
17   int i, j, p;
18   for(;;) {
19     for (j=0;j<STRLEN;j++) {                //Iterate through display string
20       for (i=0;i<COLS;i++) {                //Iterate through LCD display
21         if (i+j>COLS+BLANKSPACES) {         //LCD string has reached end of display string
22           name2[i]=" ";
23         }
24         else {                                          //LCD string still within display string
25           name2[i]=name1[i+j];
26         }
27       }
28       lcd_print7(name2);
29         for (p=0;p<DELAY;p++) {                        //Delay
30         }
31     }
32   }
33   return 0;
34 }
```

Software Details - Lab 2

```
1   //Ankita Gore, Shikhar Kumar, Christina Huang
2   #define NUMCOLS 7            //Number of columns in calculator configuration
3   #define NUMROWS 6            //Number of rows in calculator configuration
4
5   int keyboard_key() {
6     int r;
7     int c;
8     for (c = 0; c < NUMCOLS; c++) { //iterate through columns
9       keyboard_column_low(c);  // change current  column to low
10      for (r = 0 ; r < NUMROWS ; r++) { // iterate through rows in current column
11        if (!keyboard_row_read(r)) {  // current row also reads low
12          keyboard_column_high(c);
13          return (c*10)+r;            //return array position of key
14        }
15      }
16      keyboard_column_high(c);  //reset column back to high
17    }
18    return -1;
19  }
20
21  int main() {
22
23    lcd_init();
24    keyboard_init();
25    for (;;) {
26      int x = keyboard_key();
27
28      if (x==-1) {
29        lcd_print7("No key");
30      }
31      else {
32        lcd_print7("KEY ");
33        lcd_put_char7('0'+(x/10), 4);  //place first digit of keyboard_key in fourth spot
34        lcd_put_char7('0'+(x%10), 5);  //place second digit of keyboard_key in fifth spot
35      }
36    }
37    return 0;
```

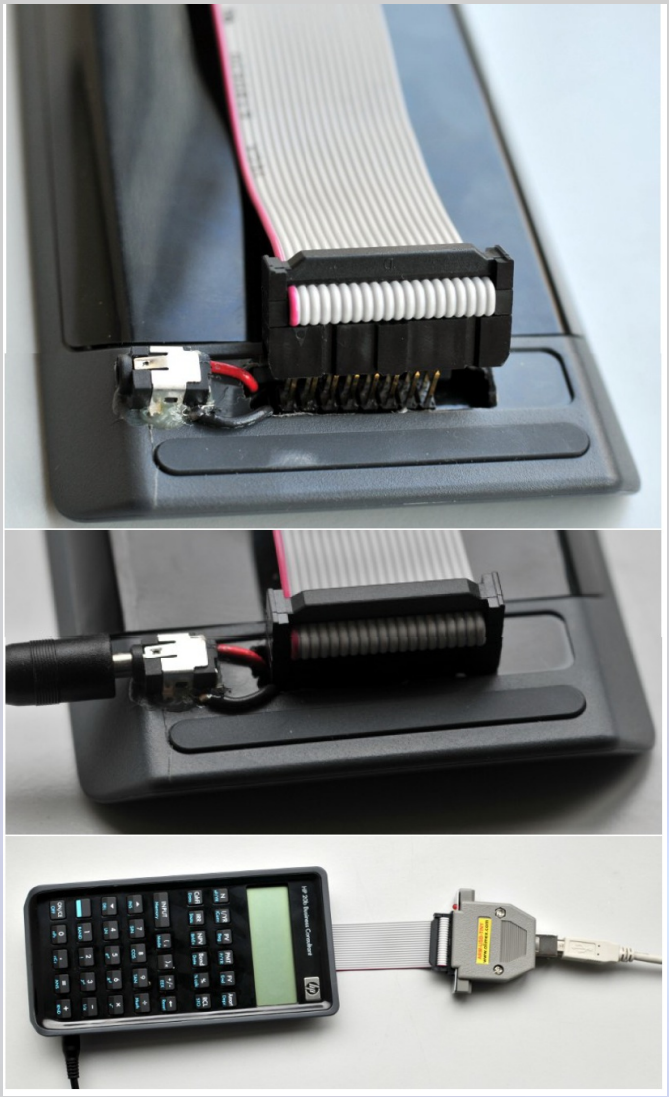
Software Details - Lab 3

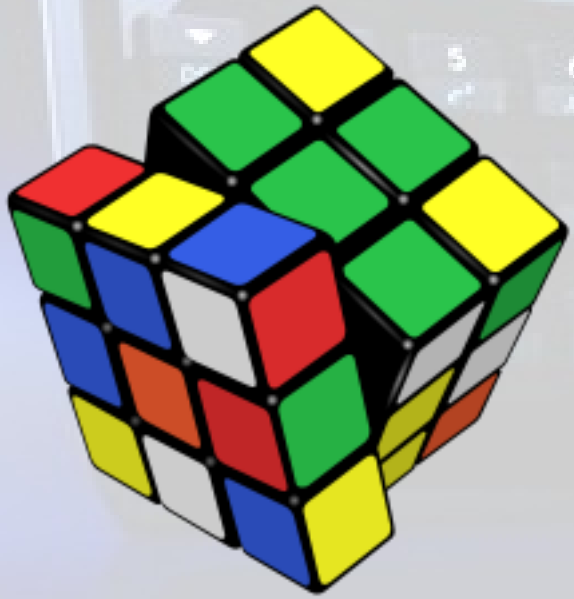
```
// Ankita Gore, Shikhar Kumar, Christina Huang
void keyboard_get_entry(struct entry *result) {
  result->number = 0;  // entry.number
  int a = -1;  // nothing is being pressed
  int numOfKeysPressed = 0;
  int sign = 1; // sign is positive
  int position = 1;

  for (;;) {
    int b = a; // b stores previous value of a
    a = keyboard_key();
    if (a>='0'  && a<='9' && b==1) {  // a is an integer and a has not been held down
      lcd_put_char7(a, position); // displays key pressed in appropriate position
      position++;
      numOfKeysPressed++;
      int num_to_add = a- '0';  // converts key pressed to int and adds to entry number
      result -> number = (result -> number)*10 + num_to_add;
    }

    if (a=='~' && b==1) {           // negative button has been pressed
      if (sign==1) {
        lcd_put_char7('-', 0);       // put negative sign
      }
      else {
        lcd_put_char7(' ', 0);      // get rid of negative sign
      }
      sign = sign*-1;                // switch sign of variable 'sign'
    }

    if (a=='/' || a=='*' || a=='+' || a=='' || a=='\r') { // operation/input key pressed
      if(numOfKeysPressed == 0) {
        result-> number = INT_MAX; // return INT_MAX if no key pressed
      }
      if (a=='/' || a=='*' || a=='+' || a=='-') {
        lcd_put_char7(a, position);
      }
      result->operation = a;
      if (sign==-1) {
        result->number*=-1;        // Switch sign of result
      }
      return; // if operation/input pressed, break out of infinite for loop
    }
  }
}
```

Software Details - Lab 4

```
1   //Ankita Gore, Shikhar Kumar, Christina Huang
2
3   #include "AT91SAM7L128.h"
4   #include "lcd.h"
5   #include "keyboard.h"
6
7   int main() {
8
9    struct entry entry;
10  // Disable the watchdog timer
11  *AT91C_WDTC_WDMR = AT91C_WDTC_WDDIS;
12
13  lcd_init();
14  keyboard_init();
15  int stack[10];              //Declare stack
16  int numOfTermsInStack = 0;     // Keeps track of how many terms in track
17  lcd_put_char7('0',11);                //Start off by displaying 0
18
19  for(;;) {
20    keyboard_get_entry(&entry);
21
22    if(entry.number != INT_MAX) {       //Add value to stack if it is valid entry number
23      numOfTermsInStack++;
24      stack[numOfTermsInStack] = entry.number;
25    }
26
27    if(entry.operation == '+' && numOfTermsInStack > 1) {  //add
28      stack[numOfTermsInStack - 1] = stack[numOfTermsInStack - 1] + stack
[numOfTermsInStack];
29      numOfTermsInStack--;            //number of terms in stack decreases by one
30    }
31    else if (entry.operation == '-' && numOfTermsInStack > 1) {  //subtract
32      stack[numOfTermsInStack - 1] = stack[numOfTermsInStack - 1] - stack
[numOfTermsInStack];
33      numOfTermsInStack--;
34    }
35    else if (entry.operation == '*' && numOfTermsInStack > 1) {  //multiply
36      stack[numOfTermsInStack - 1] = stack[numOfTermsInStack - 1] * stack
[numOfTermsInStack];
37      numOfTermsInStack--;
38    }
39    else if (entry.operation == '/' && numOfTermsInStack > 1) {    // divide
40      stack[numOfTermsInStack - 1] = stack[numOfTermsInStack - 1] / stack
[numOfTermsInStack];
44      if(stack[numOfTermsInStack] > INT_MAX
45                || stack[numOfTermsInStack] <
INT_MAX*-1) 46 {  //print error if input value too big
47        lcd_print7("ERROR!                    ");
48        numOfTermsInStack = 0;
49      }
50      else {      //print value in stack
51        lcd_print_int(stack[numOfTermsInStack]);
52      }
53
54    }
55    return 0;
56  }
```

Lessons Learned





Criticism of the Course

References

[1] Lab 1. Online http://www.cs.columbia.edu/~sedwards/classes/2011/gatewayfall/hello.pdf.

[2] Hp-20b repurposing project. Online http://www.wiki4hp.com/doku.php?id=20b:repurposing_project.

[3] Hp-20b. Online http://en.wikipedia.org/wiki/HP_20b.

[4] Lab 2. Online http://www.cs.columbia.edu/~sedwards/classes/2011/gatewayfall/keyboard.pdf.

[5] Hp-20b business consultant financial calculator manual. Online http://h10010.www1.hp.com/wwpc/pscmisc/vac/us/product_pdfs/HP_20b_Online_Manual.pdf.