

# Gateway: ENGI E1112

## CS Lab Project

Ross Basri

Ruchir Khaitan

# Design Brief

- Program and integrate new firmware for an HP 20b calculator.



# Specifications

- Final product must...

# Limitations

- HP20b calculator
- Linux Workstation
- JTAG adaptor

# Lab 1: Hello World

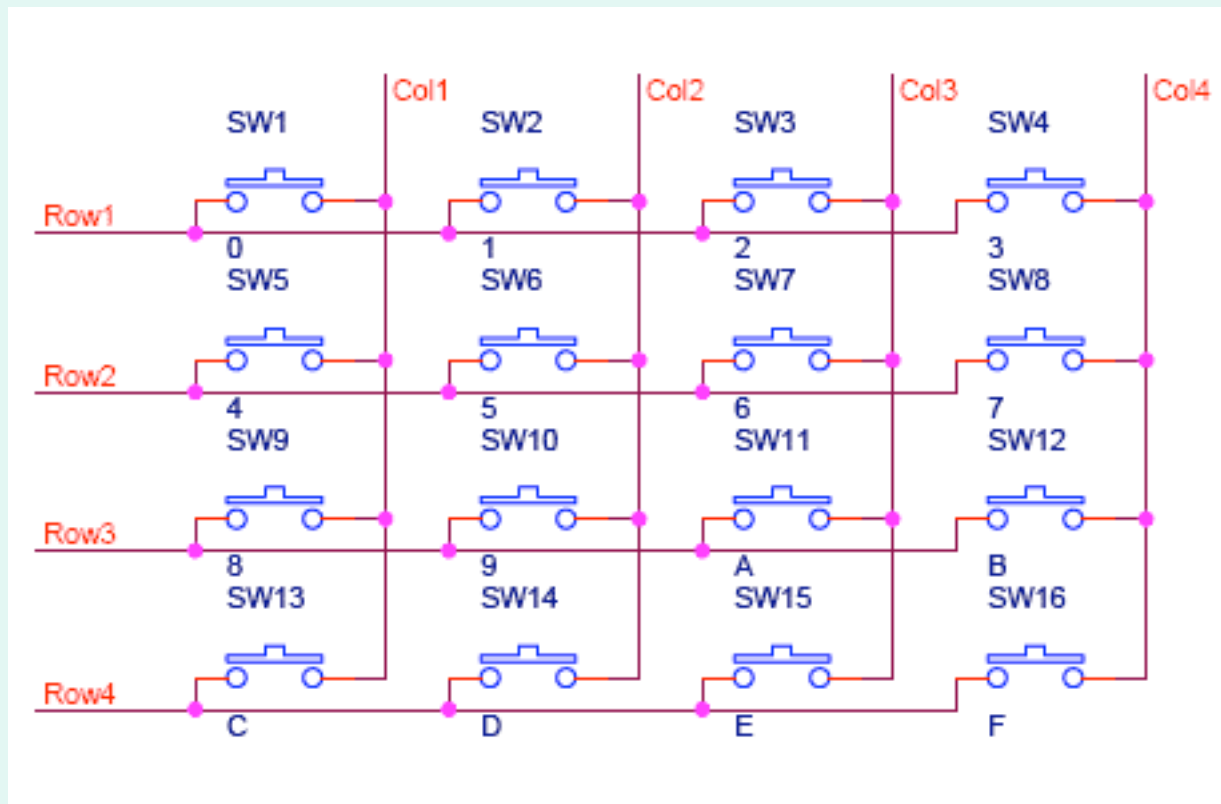
- Objective: To create a function that takes an integer argument and displays it in decimal on the calculator.

```
void lcdprint(int input){
    if(input == 0){
        lcd_put_char7('0', NUM_COLUMNS);
        return;
    }
    int remainder, count = NUM_COLUMNS;
    int negative = input < 0 ? 1 : 0;
    int output = negative ? -1*input : input;
    while(output>0){
        remainder = output%10;
        lcd_put_char7('remainder',count--);
        output = output/10;
    }
    if(negative) lcd_put_char7(45, count); //print minus sign
}
```

# Lab 2: Listening to the Keyboard

- Objective: To write software that will read the keyboard on the HP 20b and display which key is pressed.

# Lab 2: Listening to the Keyboard





```
int returnColumn=NUM_COLUMNS; //default case
extern int keyboard_key()
{
    int column, row;
    for(column=0; column<NUM_COLUMNS; column++){
        keyboard_column_low(column);
        for(row=0; row<NUM_ROWS; row++){
            if(!keyboard_row_read(row))
                return row;
            returnColumn = column;
        }
        keyboard_column_high(column);
    }
    return NUM_ROWS; //default case
}

extern int getColumn(){ //accessor for which column was pressed
    return returnColumn;
}
```

# Lab 3: Entering and Displaying Numbers

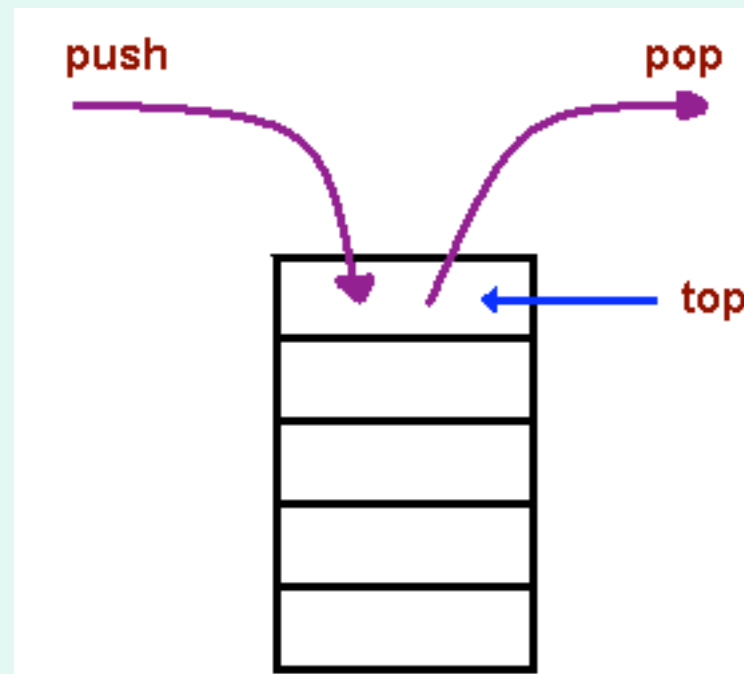
- Objective: To write code that will let the user enter and edit numbers.

```
void keyboard_get_entry(struct entry *result)
{
    int key;
    unsigned int num = INT_MAX;
    for (;;) {
        while (keyboard_key()); //wait until no key is pressed
        while (!(key = keyboard_key())); //wait until key is pressed
        if (key >= '0' && key <= '9') {
            if (num == INT_MAX) num = 0;
            if (num < 1000000000)
                num = num * 10 + (key - '0');
        }
        else if (key == '\r' || key == '+' || key == '-' || key == '*' || key == '/') {
            result->number = num;
            result->operation = key;
            return;
        }
        lcd_print_int(num);
    }
}
```

# Lab 4: An RPN Calculator

- Objective: To create a functioning, reverse-polish notation calculator.

# Lab 4: An RPN Calculator



...

```
if(current > STACK_SIZE-1) lcd_print7("OVERFLOW"); //Handle overflow
else{
    if(current < 0) lcd_print7("UNDERFLOW"); //Handle underflow
    else lcd_print_int(popped); //Print 0 in case of clear
}
}
else if(entry.operation == '\r') stack[current++]=entry.number;
else{
    popped = stack[--current];
    if(entry.number == INT_MAX) popped2=stack[--current]; //no number
pressed, only operation, i.e. 5 3 4 + ±
    else popped2=entry.number; //number and operation given
    if(entry.operation == '+') result=popped+popped2;
    if(entry.operation == '-') result=popped-popped2;
    if(entry.operation == '*') result=popped*popped2;
    if(entry.operation == '/') result=popped/popped2;
    stack[current++]=result;
    lcd_print_int(result);
}
```

To Conclude...