```
1  //Ændringer af Lucas Patrick Hayes (L 2d2 14) og Kristoffer Wienmann Biehl Christiansen (L 2d2 13)
2
3  // Active PDE program version.
4  // 8-bit Binary to Unsigned integer number
5
6
7  Bit[] bits = new Bit[8];    // array with room for 8 on/off Bit instances
8  int decimal = 0;
9  int binary = 0;
10 PFont font;
11
12 void setup() {
13   size(600, 300);
14   noStroke();
15   font = createFont("Arial", 48, true);    // Windows 10 have a wrong font path
16   for (int i = 0; i < bits.length; i++) {
17     bits[i] = new Bit(i);
18   }  // for
19 }  // setup
20
21 void draw() {
22   background(0);
23   for (int i = 0; i < bits.length; i++) {
24     bits[i].display();
25     fill(255);
26     int bitValue = 1 << (bits.length - i - 1);    // very fast calculation of 2^i
27     if(bitValue == 128){                    //ændring så når den specifikt skal vise 128 viser den -128 i
28 stedet
29       bitValue = -128;
30     }
31     text(bitValue, width/9 * bits[i].position - 10, 50);
```

```
32      } // for
33      fill(255);
34      textFont(font, 48);
35      textAlign(RIGHT);
36      text(nf(binary, 8), width/9*8, 180);
37      text(decimal, width/9*8, 230);
38      textAlign(LEFT);
39      text("Binært:", width/9, 180);
40      text("Decimalt:", width/9, 230);
41      textFont(font, 16);
42      fill(0, 255, 255);  // Cyan text
43      text("Klik en bit for at tænde eller sluk for en bit i en 8-bit 2's complement bit streng.", width/22,
44    25);    //ændring i tekst så den passer til den nuværende funktion
45    } // draw
46
47    void keyReleased() {
48      decimal = 0;
49      binary = 0;
50      for (int i = 0; i < bits.length; i++) {
51        bits[i].updateKey();
52        if(i == 0){                    //Hvis bitten er bitten mest til venstre
53          decimal -= bits[i].value;       //skal den trækkes fra den totale værdi
54        }
55        else{                          //ellers bare læg til som normalt
56        decimal += bits[i].value;
57        }
58        binary += bits[i].digit;
59      } // for
60    } // keyReleased
61
62    void mouseReleased() {
```

```
63      decimal = 0;
64      binary = 0;
65      for (int i = 0; i < bits.length; i++) {
66        bits[i].updateMouse();
67        if(i == 0){                    //Hvis bitten er bitten mest til venstre
68          decimal -= bits[i].value;       //skal den trækkes fra den totale værdi
69        }
70        else{                          //ellers bare læg til som normalt
71        decimal += bits[i].value;
72        }
73        binary += bits[i].digit;
74      } // for
75    } // mouseReleased
76
77    class Bit {           // Bit object class
78      int position;
79      color colour = (55);  // Grey
80      int value = 0;
81      int digit = 0;
82
83      Bit(int pos) {
84        position = pos + 1;
85      }
86
87      void display() {
88        fill(colour);
89        ellipse(width/9 * position, 80, 50, 50);
90      }
91
92      void updateKey() {
93        if (key == position + 48) {
```

```
94          switch(colour) {
95            case (55):
96              colour = (255);
97              value = int (pow(2, 8 - position));
98              digit = int (pow(10, 8 - position));
99              break;
100           case (255):
101             colour = (55);
102             value = 0;
103             digit = 0;
104             break;
105         } // switch
106       } // if
107     } // updateKey
108
109     void updateMouse() {
110       if (onCircle(width/9 * position, 80, 50)) {
111         switch(colour) {
112           case (55):
113             colour = (255);
114             value = int (pow(2, 8 - position));    // slow calculations
115             digit = int (pow(10, 8 - position));
116             break;
117           case (255):
118             colour = (55);
119             value = 0;
120             digit = 0;
121             break;
122         } // switch
123       } // onCircle
124     } // updateMouse
```

```
125   }  // class
126
127   boolean onCircle(int x, int y, int diameter) {
128     float distX = x - mouseX;
129     float distY = y - mouseY;
130     int radius = diameter / 2;
131     if ( sqrt(sq(distX) + sq(distY)) < radius ) {
132       return true;
133     } else {
134       return false;
135     }  // if
136   }  // end onCircle
137
138   // end
```