Project Background

Over the past several years people have increasingly acquired virtual possessions. These include both possessions that are losing their material integrity (books, photos, music, movies) as well as things that have never had material form (e.g. email archives, social networking profiles, personal behavior logs). However, little is known about how people perceive, value, and form attachments to their virtual possessions.

To begin to investigate this emerging area, I built the TxTCloud Shadow Box. SMS storage is increasingly moving from local handsets to remote servers, creating the opportunity for everyday messages to collate into archives. The TxTCloud Shadow Box presents wordclouds of a user's text message archives that are associated with particular people or social groups. There are four switches, each of which a particular archive is associated with. These switches enable the user to tangibly interact with these archives by creating different combinations of worldclouds to explore their growing digital collections. I am primarily interested in how the form, presentation and presence of text message archives might shape or modify their perceived value. In other words, would the Shadow Box open a space for people to construct new value with these prosaic collections? Would it shape the ways in which people think about social relationships implicated in these archives, and perhaps how they communicate with them?

The following sections present (i) construction and parts list, (ii) documentation of the construction process, (iii) breadboard configuration, (iv) commented code, (v) photos of a use scenario, (vi) a link to a video of the prototype, and (vii) a brief reflection on future directions.

Construction

Parts list

Electronics:

- 17 inch VGA monitor
- Arduino Duemilanove
- Small Breadboard
- Alligator Clips (for connecting wires; and for connecting artifacts to switches on top of box)
- Electric Wire
- Resisters

Form building:

- Aluminum Foil (to open and close switch)
- Acrylic Cubes (covered in foil to open and close switch)
- 12 Gauge wire (to make switch tangible)
- Blackboard (to construct sides of shadowbox and internal frame)
- Frame Molding
- Glass
- Particle Board (to screw frame into and construct a mount to center VGA display)
- Cardboard (for internal supports and guides)
- Foam (mount switches in)
- Foamcore (construct switch guides)
- Hot Glue
- Wood Glue
- Magnets

Tools Used:

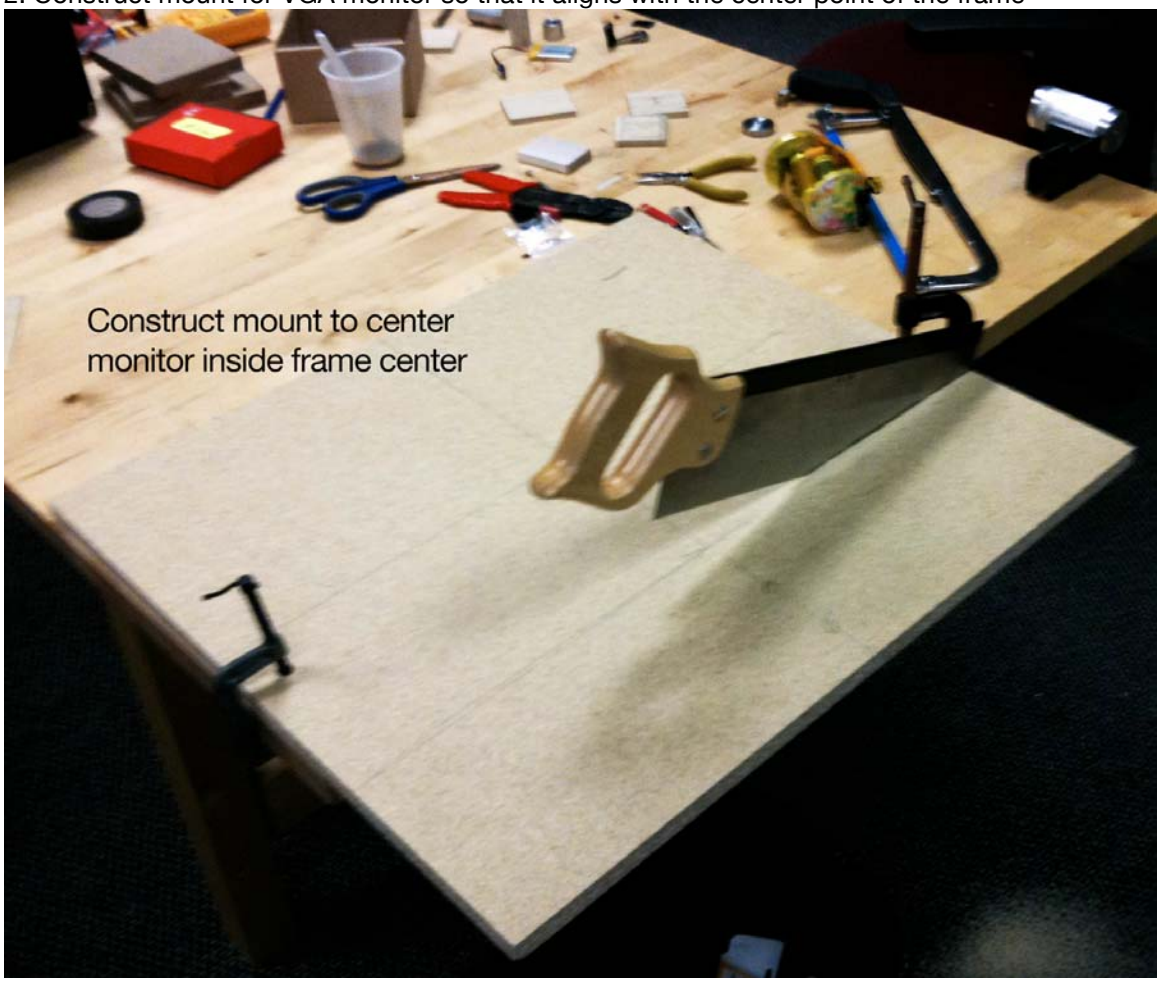
- Spring Clamps
- Drill, Drill bits
- Countersink
- Hot glue gun
- Wood saw
- Razor, Straight Edges
- Processing

Construction Process

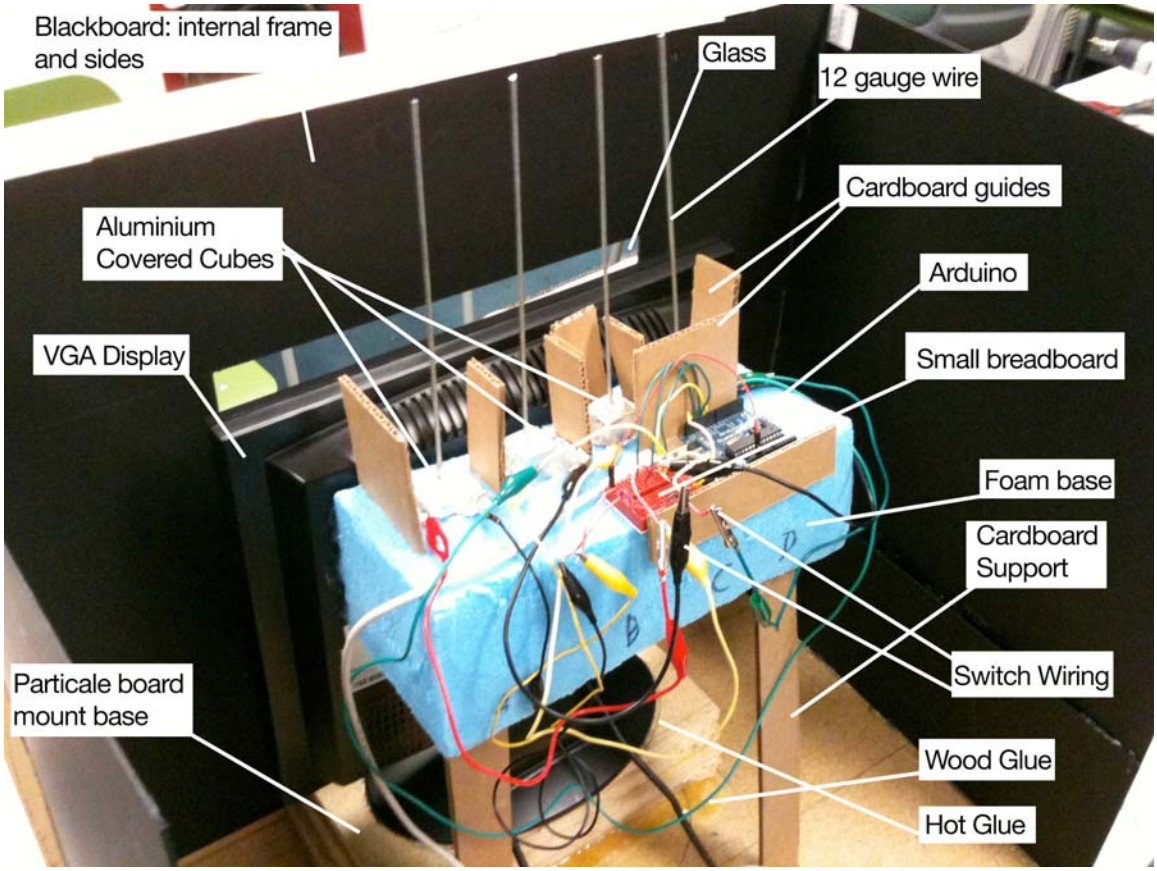
0. get the tools to mount frame to base particle board (and power drill not pictured)

1. Clamp frame to board (upside down); use drillbit, countersink, and then screwdriver (in that order) to screw frame to base board

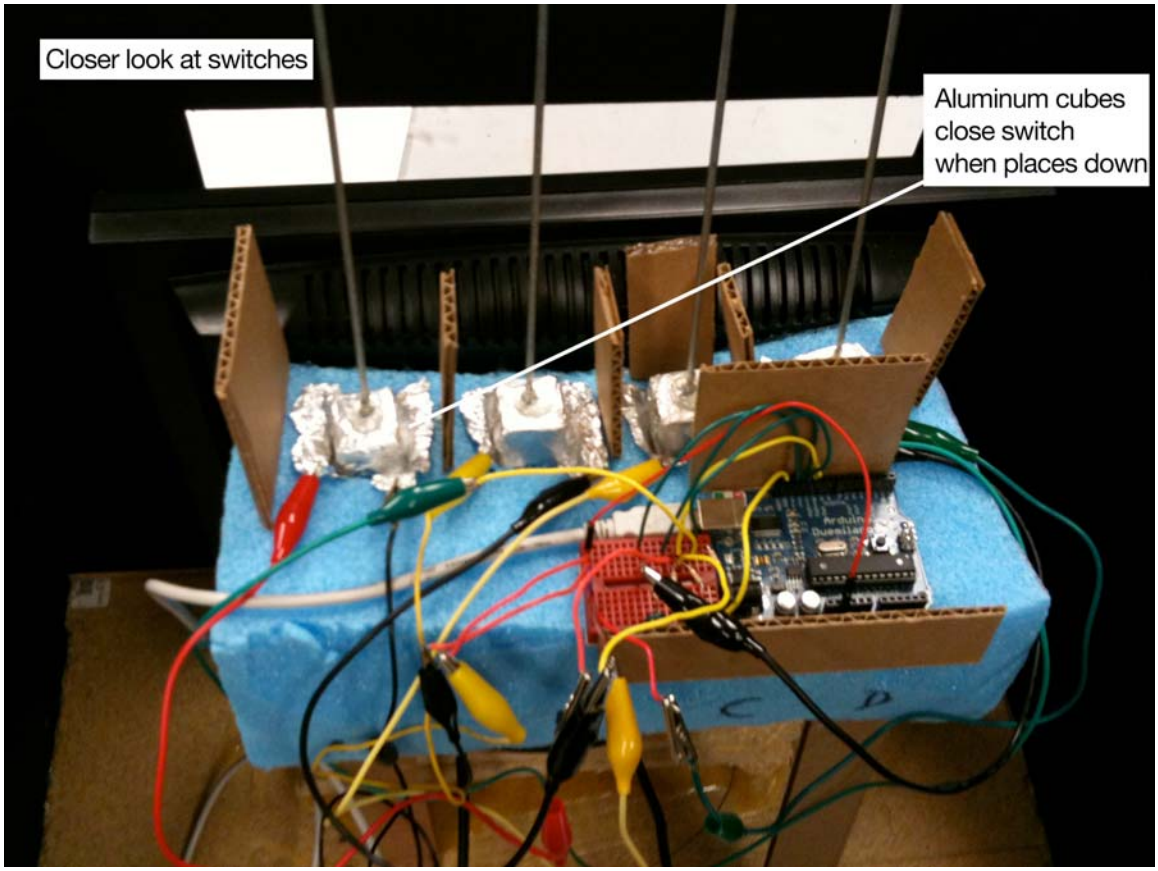
2. Construct mount for VGA monitor so that it aligns with the center point of the frame



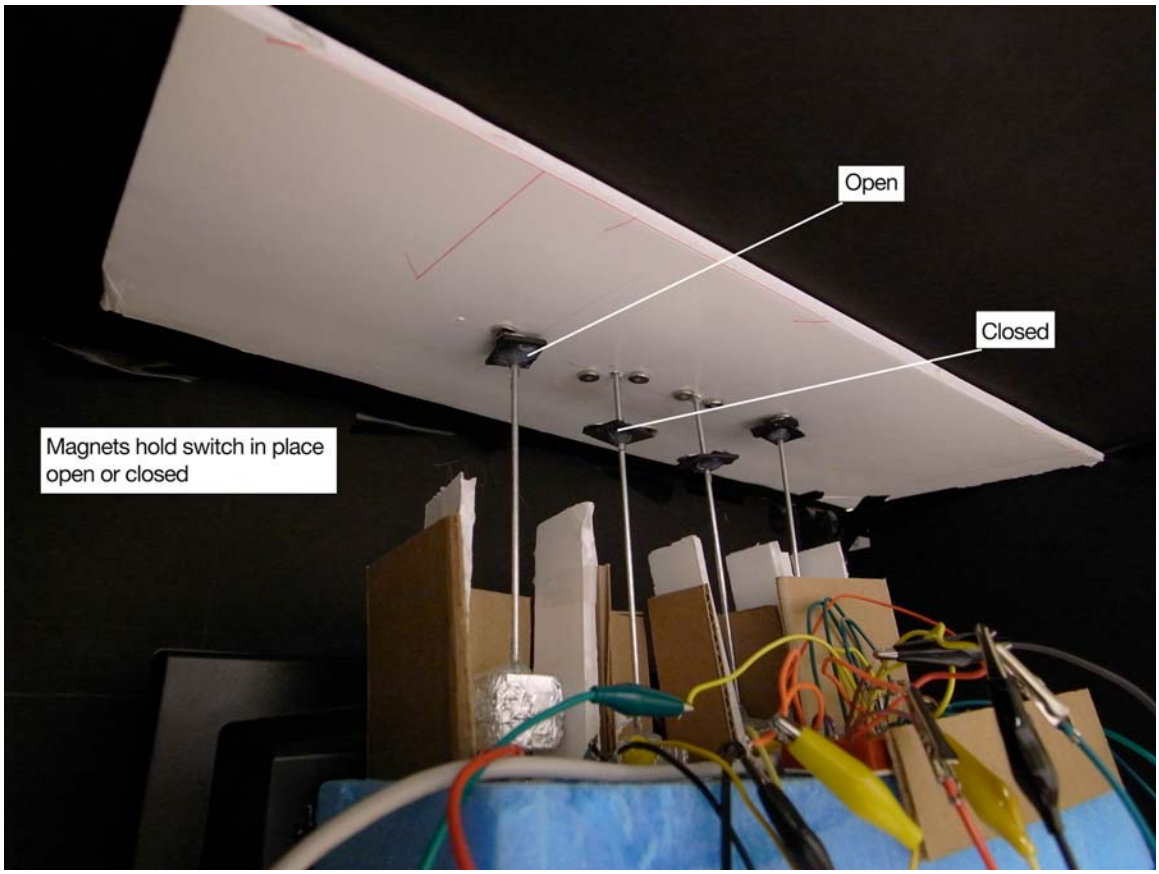
3. Create container for switches out of foam, layer each compartment with foil on each side (but not touching). Cover cubes with foil and use hot glue to attach 12 gauge wire (cut to 17cm in this case). Cut cardboard guides where needed. Cut blackboard to the dimension of the base board and external frame to enclose the components; also cut internal frame out of blackboard. Wire switches using small breadboard to arduino and each compartment with foil (in this case clips are used to construct switches; in final deployment next semester I will solder). Use wood gule to put pieces of particle board together, which form the base for the VGA monitor. Hot glue VGA monitor to particle board mount; and hot glue foam switch holder to back of VGA monitor. Construct cardboard supports for foam holder if desired.



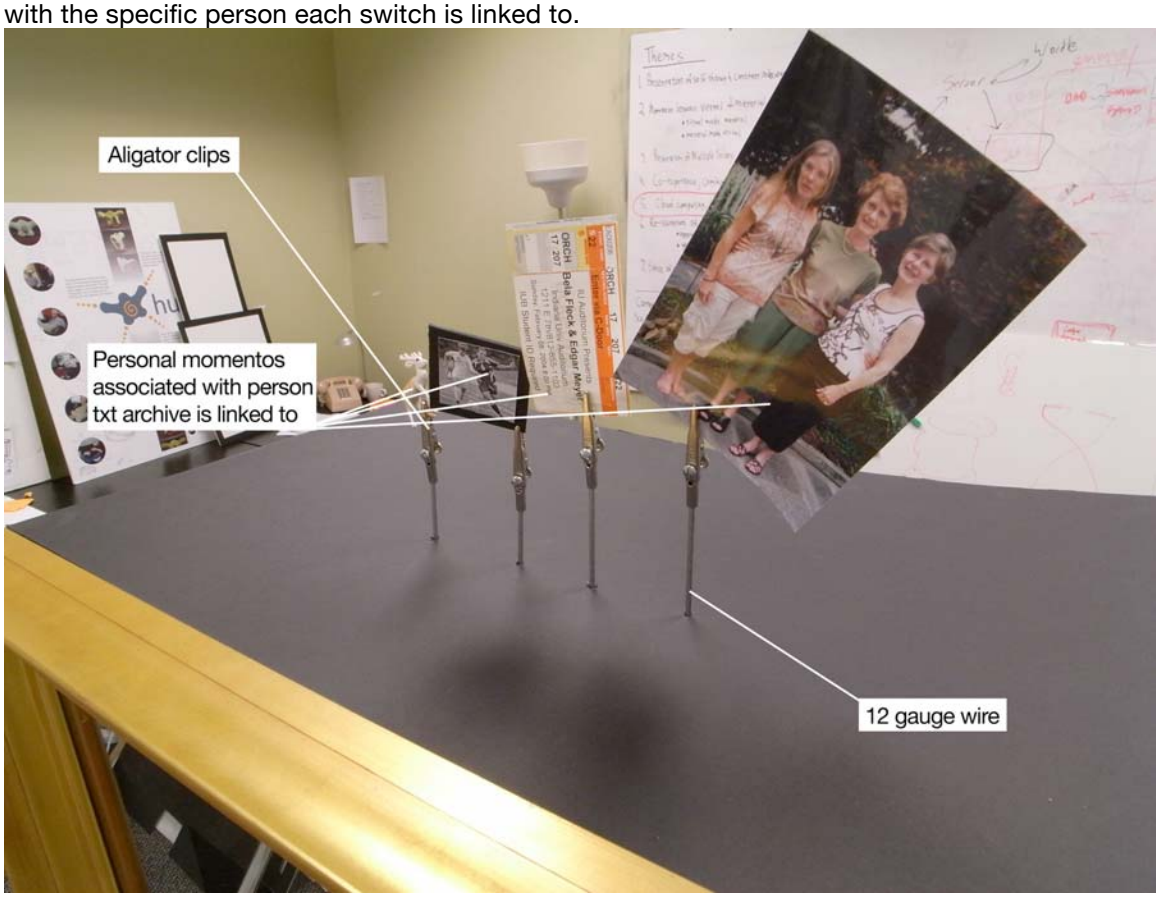
4. Not exactly a step, but this provides a closer look at the switches and how the cubes sit into the compartments (thus closing the switch). Foil set into the foam is hot glued to keep it in place.



5. Magnets are hot glued to roof of internal blackboard top in order to hold switches in open state when user is interacting with the device. This diagram illustrate open vs. closed. Holes are also drilled in blackboard roof for each 12 gauge wire switch assembly to easily move up and down.

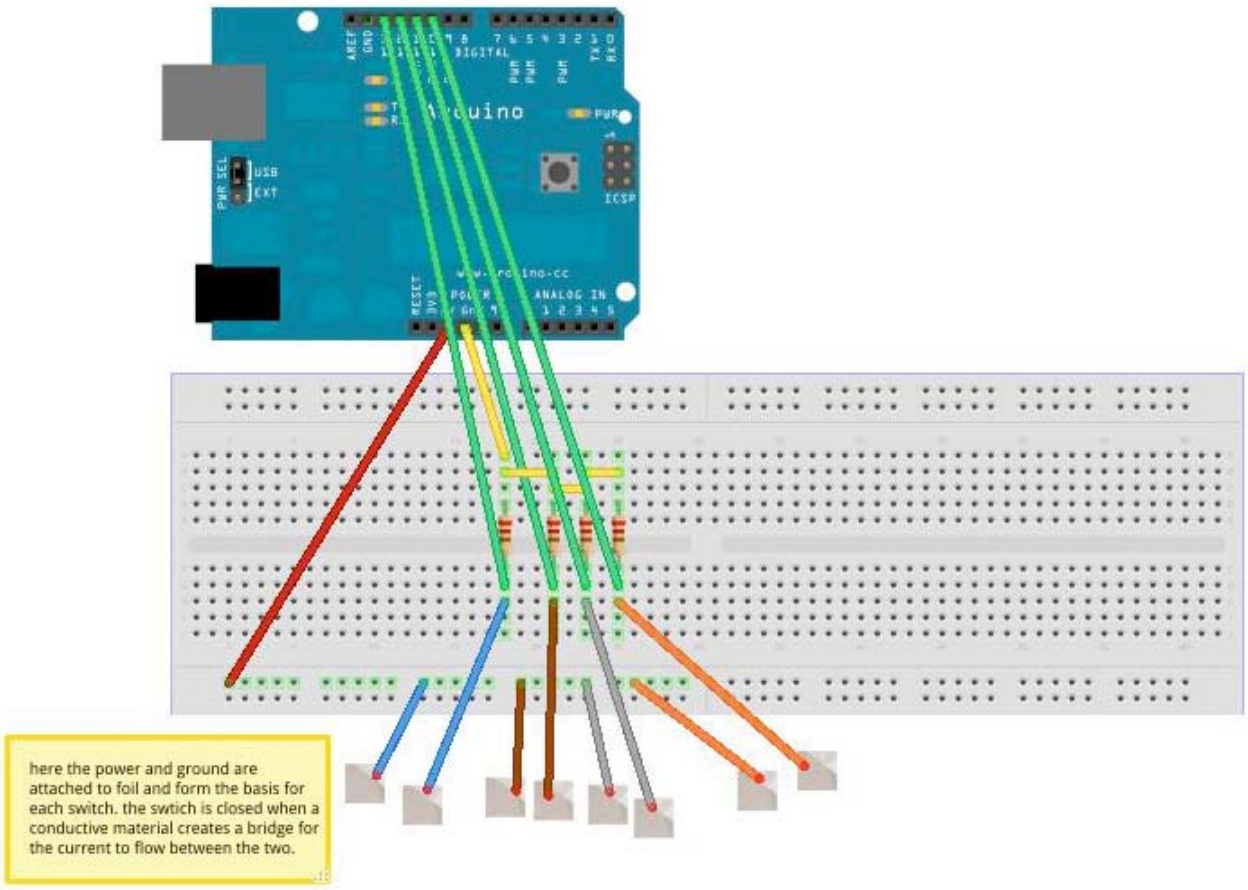


7. Affix alligator clips to top of 12 gauge wire for participants to associate personal mementos with the specific person each switch is linked to.



8. Blackboard is used to construct 3d enclosure. This view illustrates the blackboard internal frame in relation to the VGA monitor. Bottom portions of blackboard internal and external frame are hot glued to base particle board.

Circuit / Breadboard configuration



Commented Code

Arduino Code:

```
/* wodom | final project | using arduino to communicate with processing based on physical input
 */

int switchPin1 = 11;  //declare digital pin input is being read from
int switchPin2 = 10;  //declare digital pin input is being read from
int switchPin3 = 12;  //declare digital pin input is being read from
int switchPin4 = 13;  //declare digital pin input is being read from



//set digital pin as input
void setup() {

  pinMode(switchPin1, INPUT);
  pinMode(switchPin2, INPUT);
  pinMode(switchPin3, INPUT);
  pinMode(switchPin4, INPUT);
  Serial.begin(9600);
}

//print output of digital pin in serial (to test)
void loop() {
  Serial.print("a");
  Serial.print(digitalRead(11));
  Serial.print("   ");
  Serial.print("b");
  Serial.print(digitalRead(10));
  Serial.print("   ");
  Serial.print("c");
  Serial.print(digitalRead(12));
  Serial.print("   ");
  Serial.print("d");
  Serial.print(digitalRead(13));
  Serial.print("   ");
  delay(1000);
}
```

Processing Code:

```
//wodom | final project | txtcloud physical interface

 import processing.serial.*;

 Serial myPort;        // declare the serial port
 int xPos = 1;         // horizontal position of the generated screen
 PImage localPhotodef;    //allocate memory for variables that will load images on screen
 PImage photo_Nothing;
 PImage photo_aON;
 PImage photo_bON;
 PImage photo_abON;
 PImage photo_cON;
 PImage photo_acON;
 PImage photo_bcON;
 PImage photo_abcON;
 PImage photo_dON;
 PImage photo_daON;
 PImage photo_dbON;
 PImage photo_dcON;
 PImage photo_dabON;
 PImage photo_dacON;
 PImage photo_dbcON;
 PImage photo_abcdON;
 PrintWriter output; //declare print writer output to write state changes to local text file

void setup () {
```

```
 output = createWriter("testtxtfile.txt"); //creates the local text file that state changes are
written to
 photo_Nothing = loadImage("nothingon.png"); //assign specific images to variables
 photo_aON = loadImage("a.png");
 photo_bON = loadImage("b.png");
 photo_abON = loadImage("ab.png");
 photo_cON = loadImage("c.png");
 photo_acON = loadImage("ac.png");
 photo_bcON = loadImage("bc.png");
 photo_abcON = loadImage("abc.png");
 photo_dON = loadImage("d.png");
 photo_daON = loadImage("ad.png");
 photo_dbON = loadImage("bd.png");
 photo_dcON = loadImage("cd.png");
 photo_dabON = loadImage("abd.png");
 photo_dacON = loadImage("acd.png");
 photo_dbcON = loadImage("bcd.png");
 photo_abcdON = loadImage("abcd.png");

localPhotodef = loadImage("default2.jpg");

 // set the window size:
 size(1280, 1024);

 // List all the available serial ports
 println(Serial.list());
 myPort = new Serial(this, Serial.list()[0], 9600); //this line enables the arduino to communicate
with processing (reading port "0" in this case)
 myPort.bufferUntil('\n');
 // set inital background:
 background(0);
 }

//initialize pin values
int valA = 48;
int valB = 48;
int valC = 48;
int valD = 48;

boolean stateChange = false; //this state change variable is used to write to a text file when a state
change occurs

int previous = '0'; //sets previously read value to zero to do the statechange check

void draw () {

  while (myPort.available() > 0) {   //testing to read input back from arduino switch using inByte
    stateChange = false;             //initialize state change to false
    int inByte = myPort.read();      //int for port readings from arduino

//switches are read during while loop; values are temporarily saved and written to text file

  if (inByte == 'a'){
    println("prev " +previous);
    previous = valA;
    println("prev " +previous);
    while (myPort.available() <= 0) {
    }
    valA = myPort.read();
        println("val A " +valA);
    if (previous != valA){
     stateChange = true;
     println("A");
     print(previous);
    }
     if (valA == 49){
           println("a true");
    }
  }
```

```
  else if(inByte == 'b'){
    previous = valB;
    while (myPort.available() <= 0) {
    }
    valB = myPort.read();
    if (previous != valB){
     stateChange = true;
     println("B");
    }
     if (valB == 49){
       println("b true");
     }

  }

  else if(inByte == 'c'){
    previous = valC;
    while (myPort.available() <= 0) {
    }
    valC = myPort.read();
    if (previous != valC){
     stateChange = true;
     println("C");
    }
     if (valC == 49){
       println("c true");
     }

  }

  else if(inByte == 'd'){
    previous = valD;
    while (myPort.available() <= 0) {
    }
    valD = myPort.read();
    if (previous != valD){
      stateChange = true;
      println("D");

    }
     if (valD == 49){
       println("d true");
     }

  }


//if there is a statechange, then write the timestamp and exact change to a local text file
if (stateChange == true){
  output.println(year() + "-" + month() + "-" + day() + "-" + hour() + "-" + minute() + "-" +
second());
  output.println(valA + "\t" + valB + "\t" + valC + "\t" + valD + "\t" + "----END---");
  output.flush();
}

//logic to determine which image to pull and display depending on combination of switches on/off
  if (valA == 49 && valB == 49 && valC == 49 && valD == 49){
    image(photo_Nothing, 0, 0);
  }
  else if (valA == 48 && valB == 49 && valC == 49 && valD == 49){
    image(photo_aON, 0, 0);
  }
  else if (valA == 49 && valB == 48 && valC == 49 && valD == 49){
    image(photo_bON, 0, 0);
  }
  else if (valA == 48 && valB == 48 && valC == 49 && valD == 49){
    image(photo_abON, 0, 0);
  }
```

```
  else if (valA == 49 && valB == 49 && valC == 48 && valD == 49){
    image(photo_cON, 0, 0);
  }
  else if (valA == 48 && valB == 49 && valC == 48 && valD == 49){
    image(photo_acON, 0, 0);
  }
  else if (valA == 49 && valB == 48 && valC == 48 && valD == 49){
    image(photo_bcON, 0, 0);
  }
  else if (valA == 48 && valB == 48 && valC == 48 && valD == 49){
    image(photo_abcON, 0, 0);
  }
  else if (valA == 49 && valB == 49 && valC == 49 && valD == 48){
    image(photo_dON, 0, 0);
  }
  else if (valA == 48 && valB == 49 && valC == 49 && valD == 48){
    image(photo_daON, 0, 0);
  }
  else if (valA == 49 && valB == 48 && valC == 49 && valD == 48){
    image(photo_dbON, 0, 0);
  }
  else if (valA == 49 && valB == 49 && valC == 48 && valD == 48){
    image(photo_dcON, 0, 0);
  }
  else if (valA == 48 && valB == 48 && valC == 49 && valD == 48){
    image(photo_dabON, 0, 0);
  }
  else if (valA == 48 && valB == 49 && valC == 48 && valD == 48){
    image(photo_dacON, 0, 0);
  }
  else if (valA == 49 && valB == 48 && valC == 48 && valD == 48){
    image(photo_dbcON, 0, 0);
  }
  else if (valA == 48 && valB == 48 && valC == 48 && valD == 48){
    image(photo_abcdON, 0, 0);
  }


}

}
```

Sample screen image

This figure illustrates a textcloud that was produced from a user's text message archive that is specific to their soccer team. The size indicates frequency of terms; in this design, color is merely cosmetic and does not indicate additional information.

Use scenario

This series of photo illustrates a scenario of how and where the TxTCloud Shadow box prototype would be used.

The Shadow box has become an integrated part of this teenager's bedroom.

After returning home from school, the user glances at the default screen, which conveys a wordcloud of his most recent outgoing messages.

He then pulls up on the Mom switch to change the visualization.

He begins trying out different combinations.

He then takes a few moments to reflect on the different themes emerging among his groups.

The Shadow Box lives among his other possessions in the room.


Spirit
Great
Job
tonight

Video

TxTCloud Shadow Box:
http://www.youtube.com/watch?v=JxjRsoOtIU0

Future Explorations

In the future, I am planning to build a few more prototypes at roughly 1/3 the scale using a tablet computer to serve as the monitor (and to run the arduino). I am also planning to rapid prototype the internal case that holds the switches as this will likely make the design more robust for field deployment (in addition to soldering the internal components together). Below you'll find images from the solidworks file. These prototypes will be used in field deployments next semester with Teenagers that send/receive >100 text messages daily. We are aiming to submit this study to Ubicomp 2011.

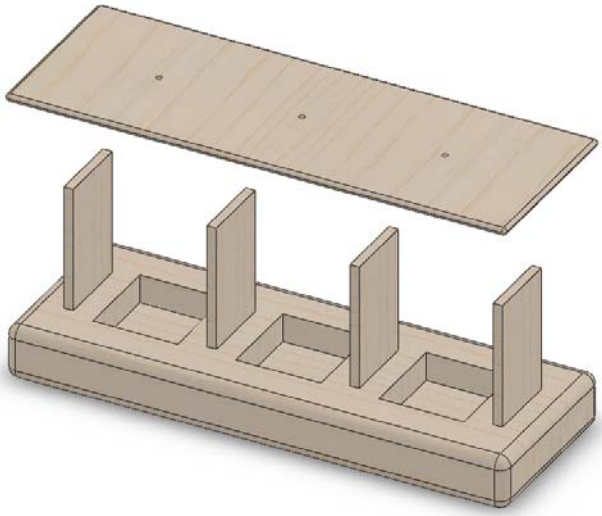


Figure 1. Top view of case and detached part to hold 12 gauge wire for photo clips.

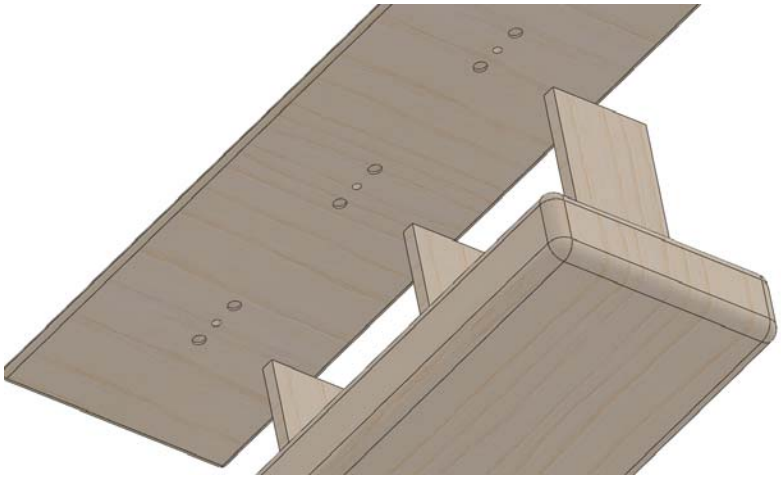


Figure 2. Partial extruded cuts on the underside to more easily affix magnets.