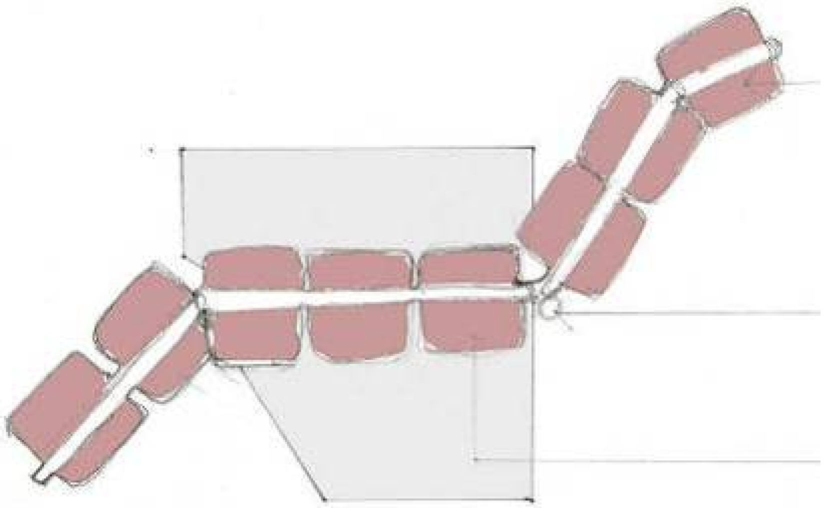
ARCH ROBOTICS
Henrique Houayek, Ivan Siles

Interactive Inflatable Furniture

IIF

Video: http://www.youtube.com/watch?v=kuf2Vn5imc0

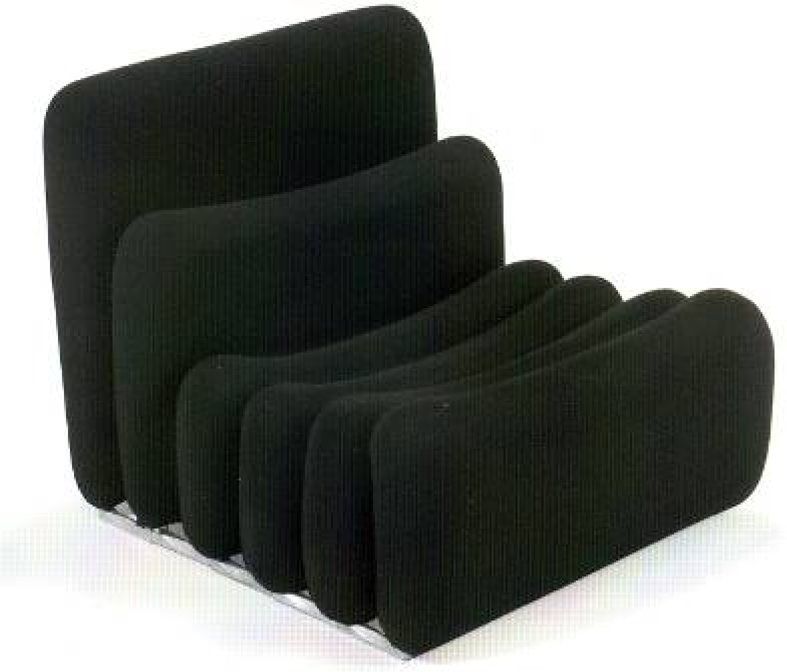


Abstract:

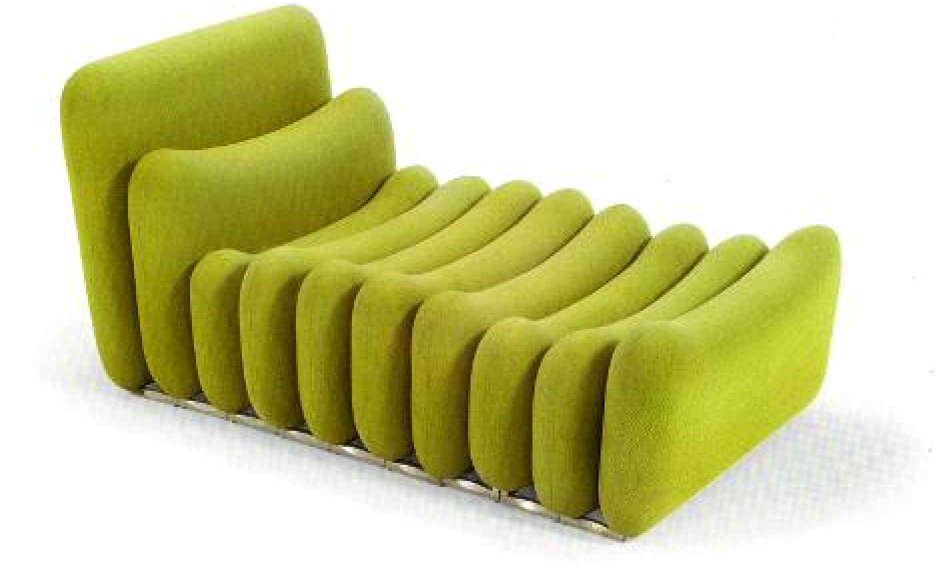
Increasing in-life time expectancy and an increasingly digital society mark two important contemporary life factors. High technology is becoming pervasive in most of our living spaces. However, domestic environments today remain essentially low-tech and conventional, neglecting human conditions for aging in place. Increasing in-life time expectancy requires special care and different spatial conditions. For most senior citizens, mundane activities such as sitting, sleeping and getting up can be a hard task.

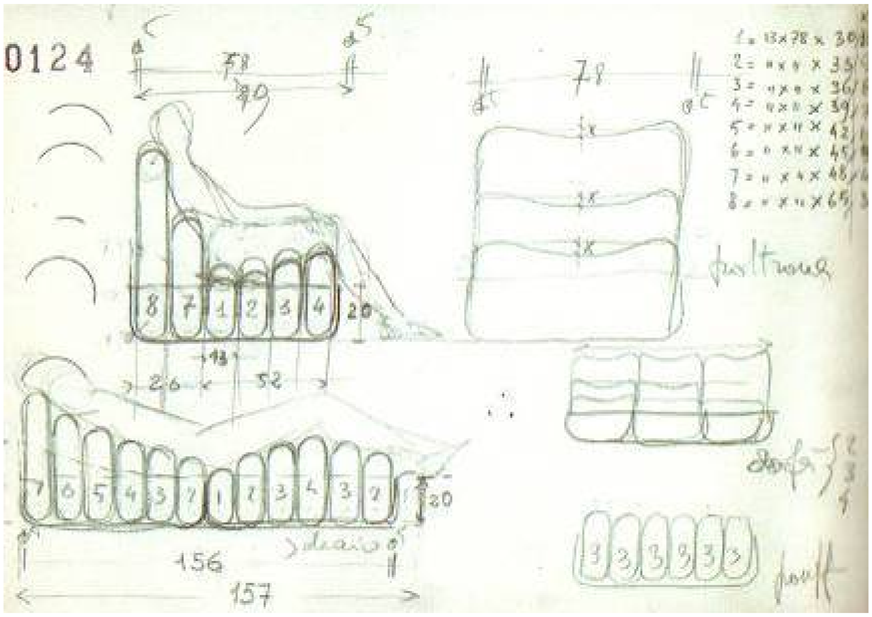
IFF is designed to increase the quality of life of both healthy elder individuals as well as persons with impaired mobility.

Design References

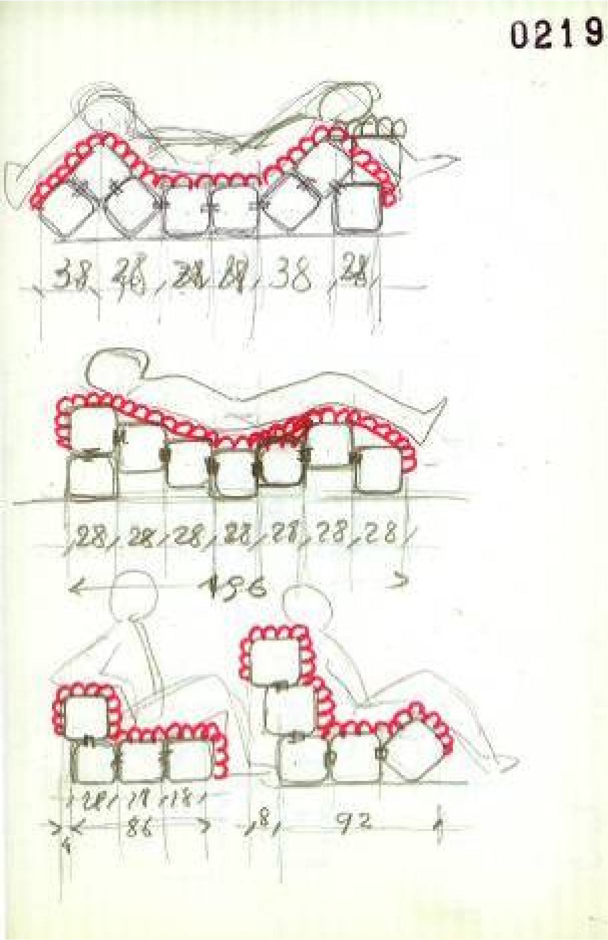


Joe Colombo, *Additional System* (Lounge Chair) 1967/68





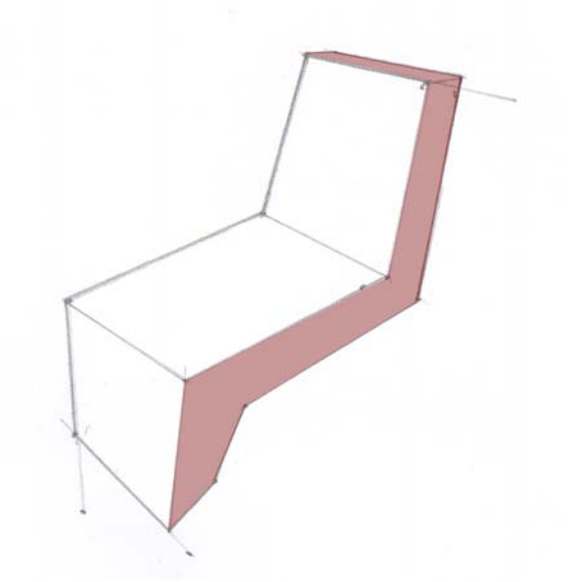
Additional System – Sketches 1967/68

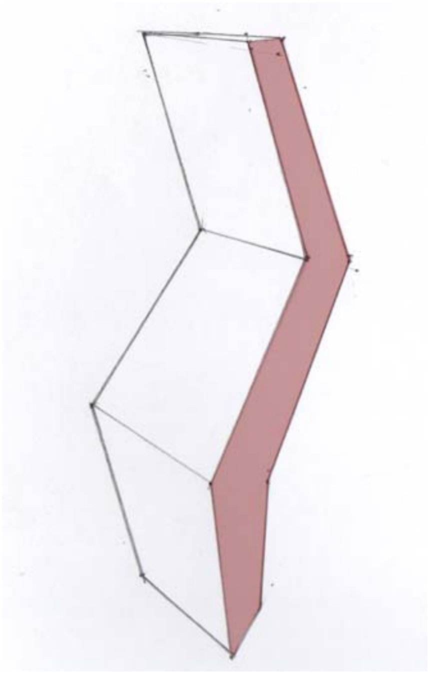


System Chair (Project for Asko) – Sketches 1971

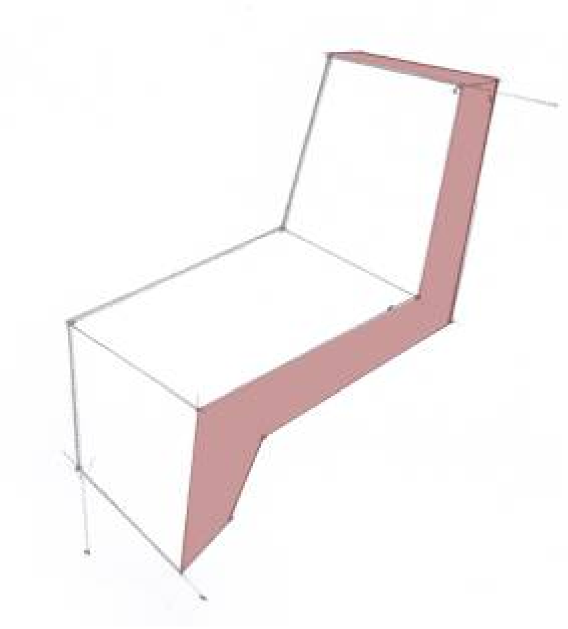
Scenario:

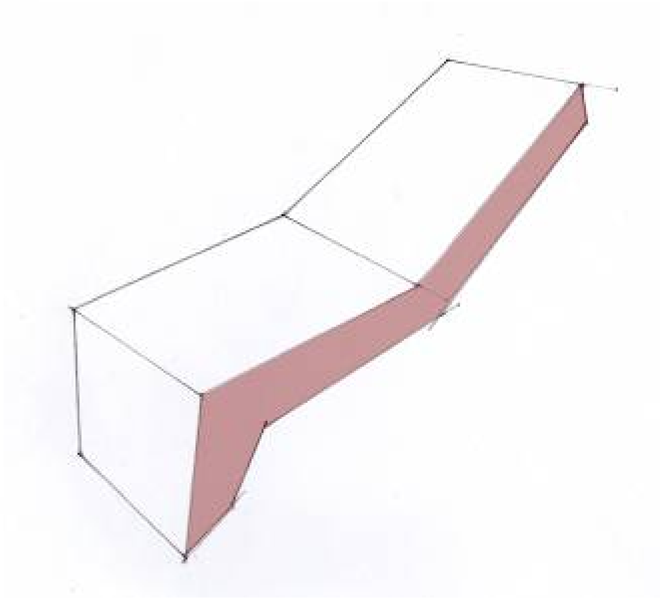
Urbanus gets close to IFF, force sensors detect his presence, and IFF adjusts its position to make it easy for him to sit.





Once Urbanus is sitting, he wants to watch television comfortably. By moving his body, an accelerometer will make IFF change form slightly. Due to changes in the body position – achieved by injecting air into the cushions and changing the angles of the furniture – a person can sit or lie comfortably for long stretches.

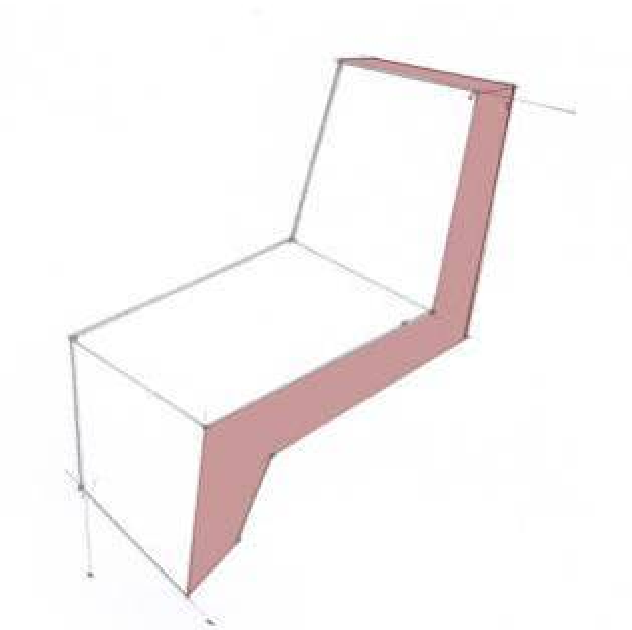


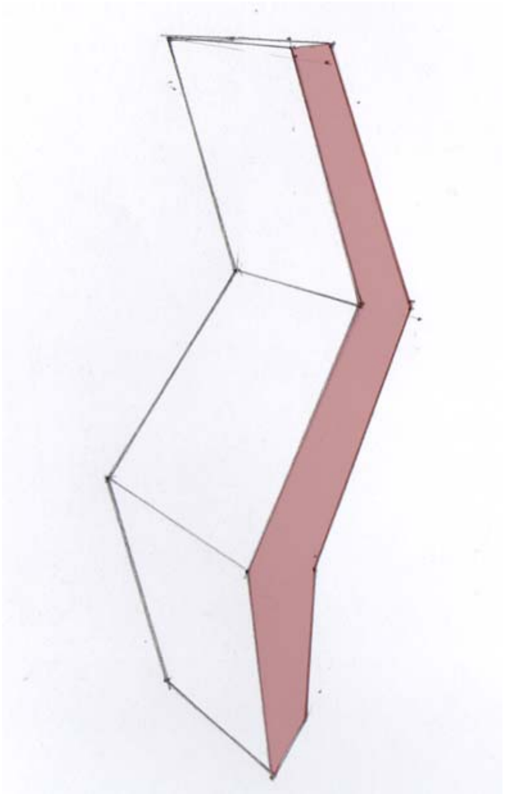


IFF can adjust to both sitting and sleeping mode.

Sensors detect head position and change the shape to sleeping mode if Urbanus relaxes the head and falls asleep, IFF sends information to the room, and the lights go off.

Once Urbanus awakes, IFF will turn the lights on, and return its shape to a sitting position.



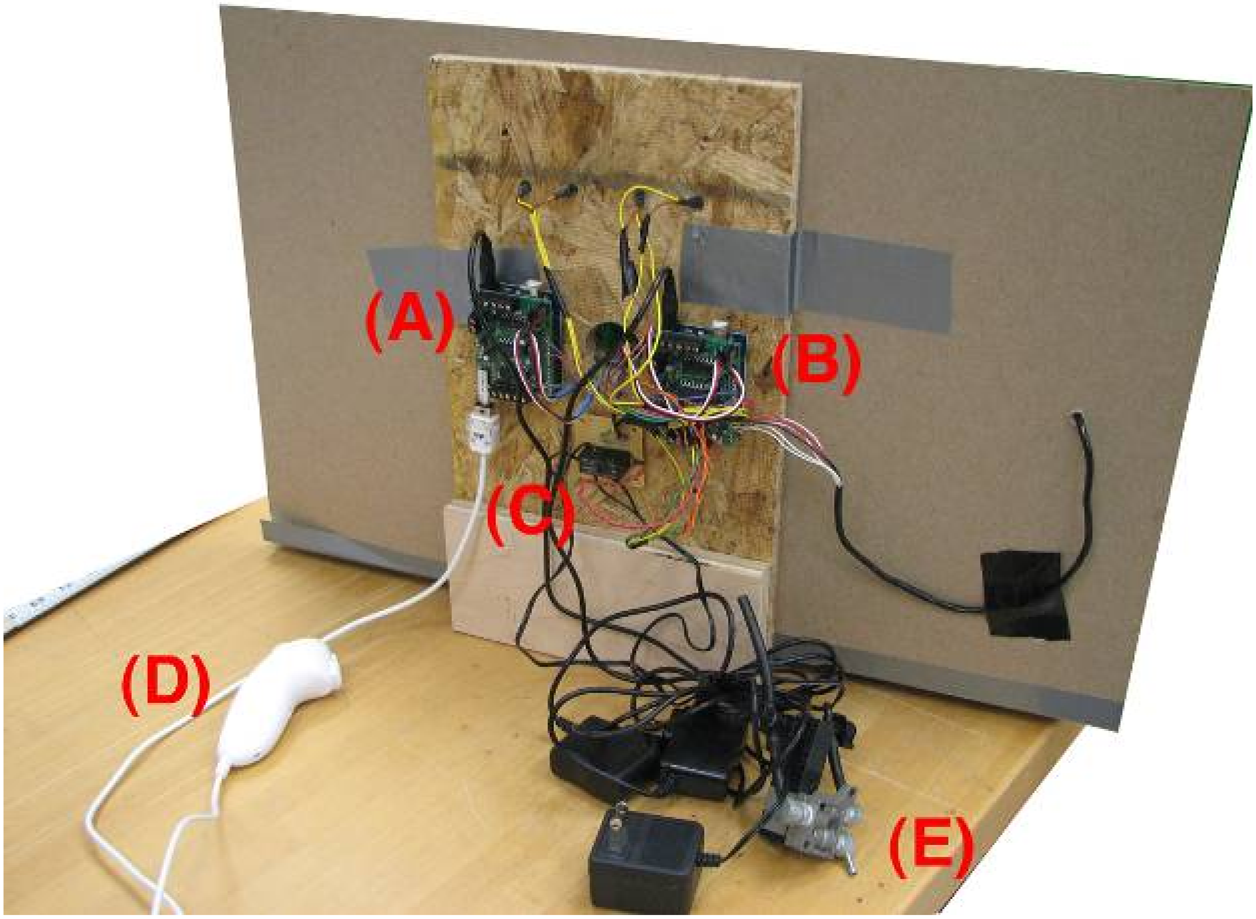


When Urbanus decides to get up, he will put his feet on the ground, and IFF will automatically adjust to standing position.

IFF can help senior citizens with locomotive problems by adapting its shape to person’s habits and needs.

Hardware components:

1. Force sensor that detects the presence of the sensor next to IFF.
2. IR sensor detects the position of the head.
3. Two servomotors adjust the shape of IFF.
4. An air valve will control the flow of air from a CO2 cartridge into the cushions in IFF.
5. A Wii nunchuck detects the position of the person.
6. A Series of LED’s work as a lighting information system.



The picture above shows the back of IFF with the main components:

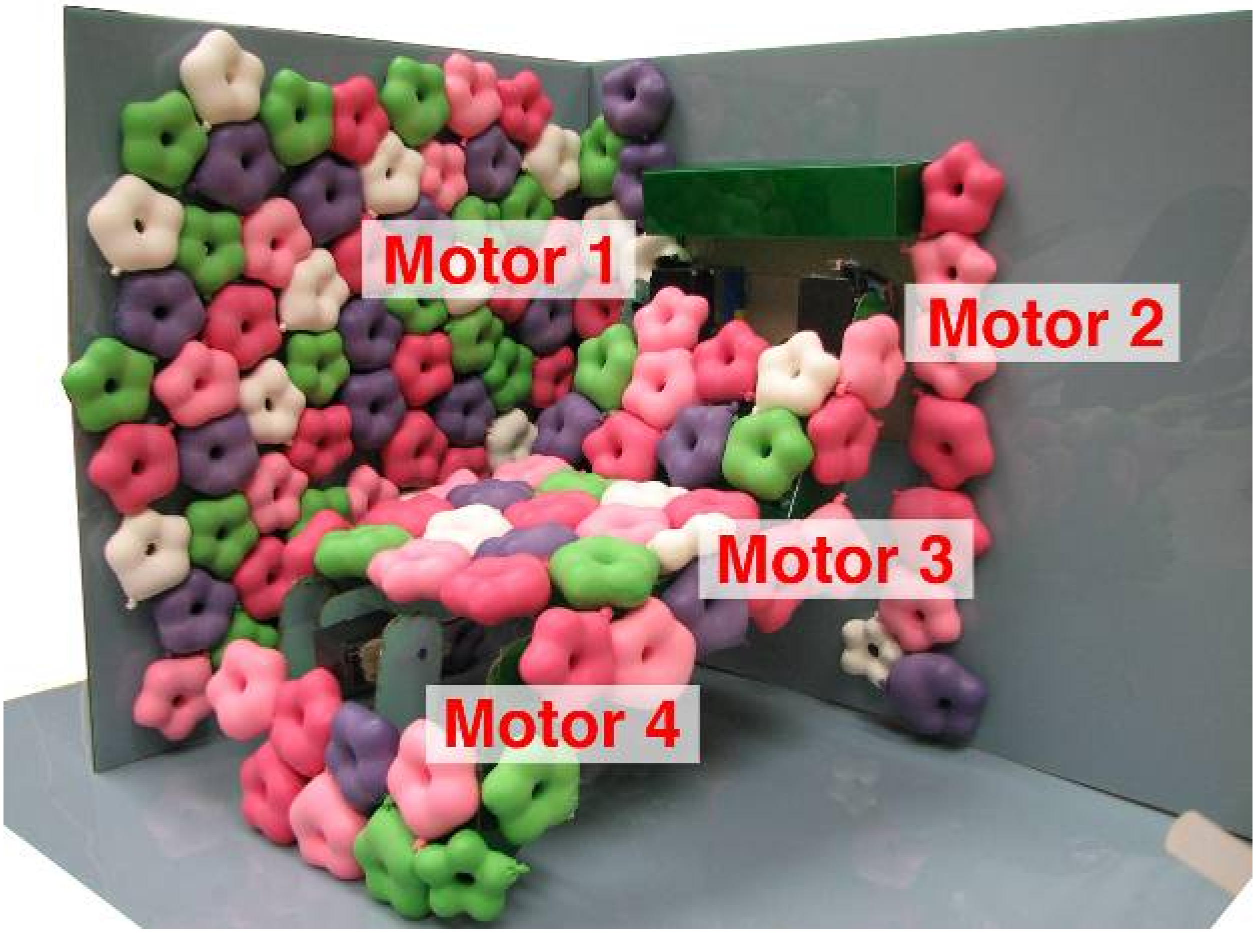
(A) Arduino 1: Controls Motors 3 and 4, and the Wii Nunchuck.

(B) Arduino 2: Controls Motors 1 and 2, LED’s, force sensor, and valve.

(C) Relay that actuates the solenoid valve.

(D) Wii Nunchuck (accelerometer).

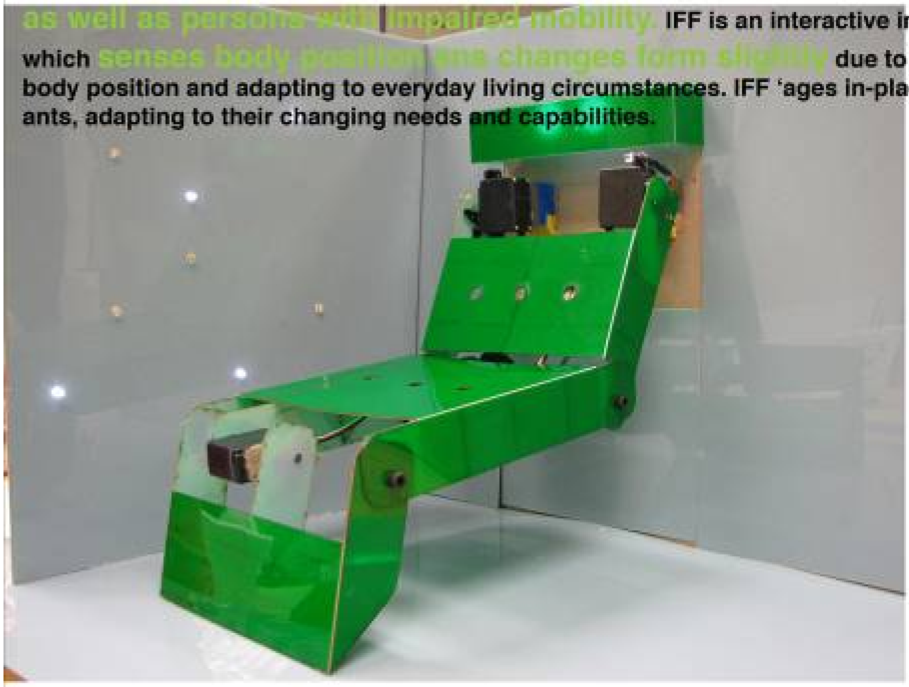
(E) Power adapters and solenoid valve.

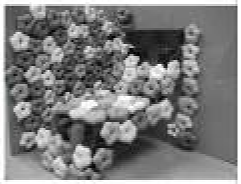

Motor 1
Motor 2
Motor 3
Motor 4

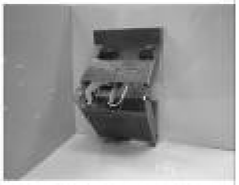
ARCH ROBOTICS
Henrique Houayek, Ivan Siles

Increasing in life time expectancy and anincreasingly digital society. mark ttwo important contemporary life factors. High technology is becoming pervasive in most of our living spaces. However, domestic environments today remain essentially low–tech and conventional, neglecting human conditions for aging in place. Increasing in life time expectancy requires special care and different spatial conditions. For most senior citizens, mundane activities such as sitting, sleeping and getting up can be hard task.

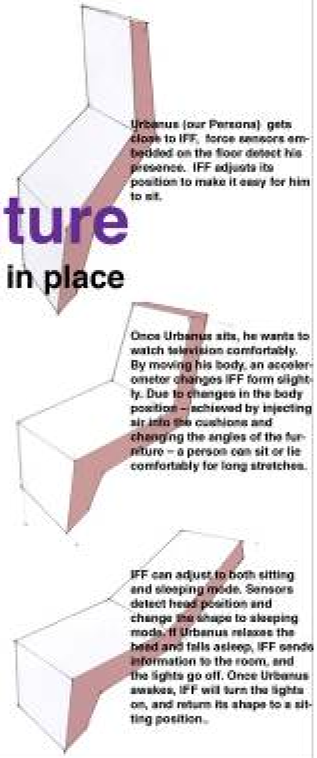
IFF is designed to increase the quality of life of both healthy elder individuals as well as persons with impaired mobility. IFF is an interactive inflatable furniture which senses body position and changes form slightly due to changes in the body position and adapting to everyday living circumstances. IFF ‘ages in-place’ with its inhabitants, adapting to their changing needs and capabilities.







Possible scenarios:



Urbanus (our Persona) gets close to IFF, force sensors embedded on the floor detect his presence. IFF adjusts its position to make it easy for him to sit.

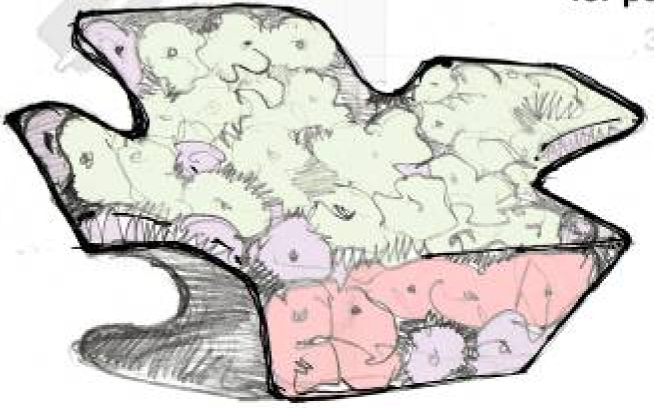
Once Urbanus sits, he wants to watch television comfortably. By moving his body, an accelerometer changes IFF form slightly. Due to changes in the body position – achieved by injecting air into the cushions and changing the angles of the furniture – a person can sit or lie comfortably for long stretches.

IFF can adjust to both sitting and sleeping mode. Sensors detect head position and change the shape to sleeping mode. If Urbanus relaxes the head and falls asleep, IFF sends information to the room, and the lights go off. Once Urbanus awakes, IFF will turn the lights on, and return its shape to a sitting position.

IFF

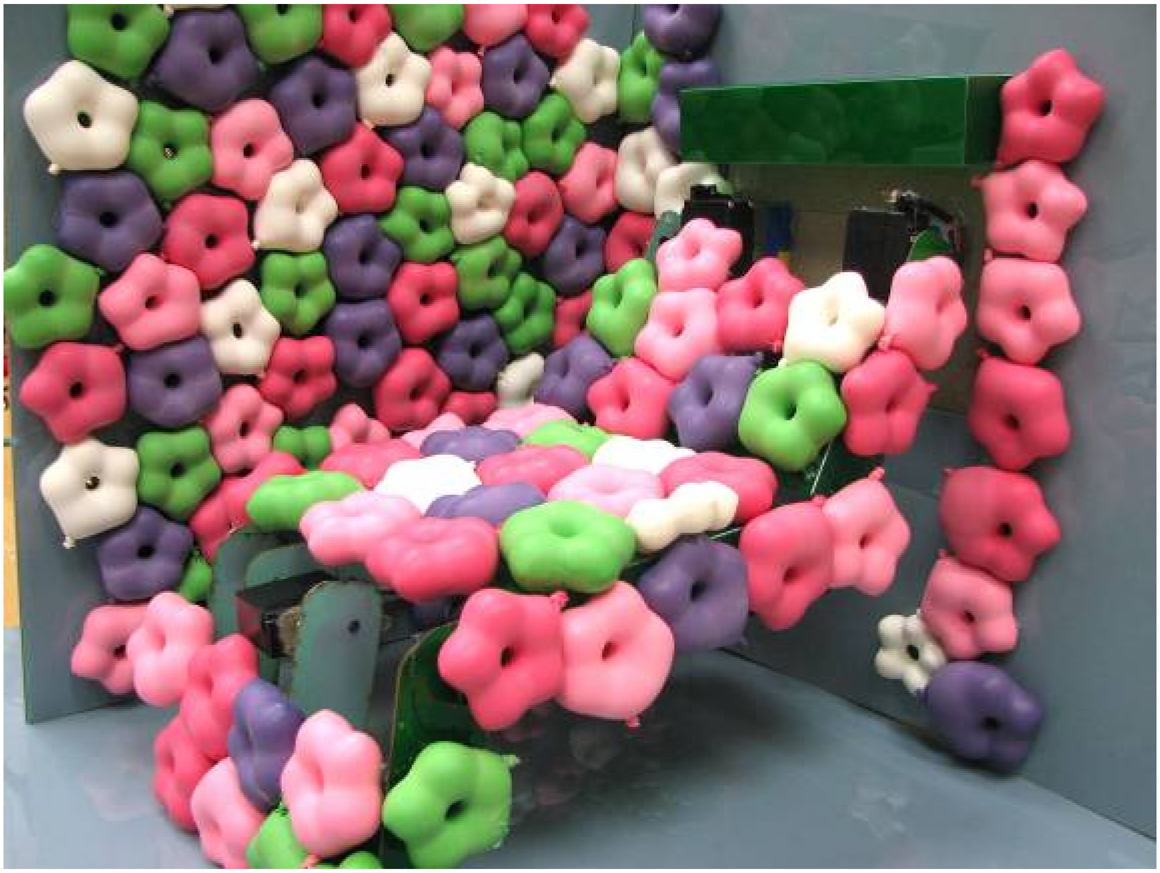
Interactive Inflatable Furniture

for people aging in place

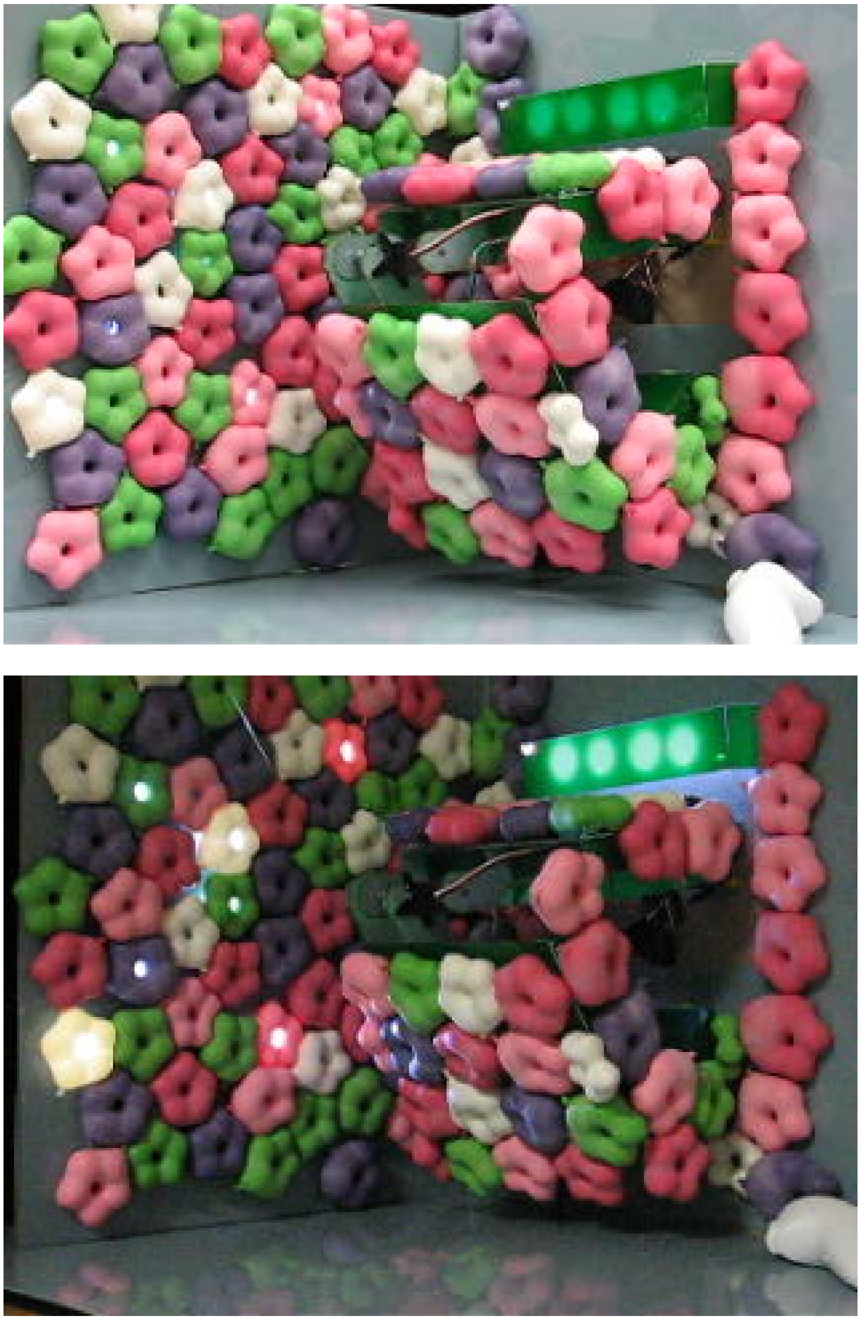


Henrique Houayek; Ivan Siles ARCH 879/ECE 893

IFF being demonstrated by Authors in Architectural Robotics Exhibition – April 22, 2009



IFF in a sitting configuration



IFF in a folded configuration

Troubleshooting:

While working on the building process of IFF, three main problems were found:

- The total weight of the device + the weight of the balloons proved to be a little too much for the servomotors. In some cases we had to “help” them with a string in order to be able to lift the whole furniture from sitting to sleeping configuration.

- The compressed CO2 bottles are hard to work with. As the air comes out the cartridge, it freezes, being really hard to keep the tubes sealing intact. Leaks appeared in the joint between the cartridge and the solenoid valve. We were not able to solve this issue given the time and resources available.

- Coordinating the two Arduino boards was quite hard, since the 4 servomotors had to be perfectly synchronized. A lot of testing was required in order to tune them until the movement was somewhat smooth.

```
//////////////////////////////////////////////////////////////////////////////////////////
//                                                                                //
//       Interactive Inflatable Furniture - Board A (New Arduino)                  //
//                                                                                //
//       Henrique Houayek and Ivan Siles                                             //
//                                                                                //
//     Project #3 - Architectural Robotics (Arch879/ECE893)                        //
//                                                                                //
//       Spring 2009                                                                   //
//       Clemson University, Clemson, SC                                               //
//                                                                             //
//////////////////////////////////////////////////////////////////////////////////////////

#include <Servo.h>

// Declare all the IN/OUT pin variables

// Inputs
int Force_S = 0;        // Force sensor connected to Analog Input 0
int Head_S = 12;         // Head IR sensor to Digital Input 0
int Comm3 = 3;           // Pin 3 to communicate with Arduino B (Input)

//Outputs
int Valve = 8;           // Solenoid valve to Digital Output 8
int LED1 = 5;
int LED2 = 4;
int LED3 = 7;
int LED4 = 6;
int Comm2 = 2;           // Pin 2 to communicate with Arduino B (Output)
int Comm1 = 11;          // Pin 1 to communicate with Arduino B (Output)

//Servos
Servo S1;
Servo S2;

// Program variables

int Force = 1023;
int Head = 0;
int Sleep = 0;
int Wii = 0;
int Awake = 0;
int pos1 = 0;
int pos2 = 0;

int i = 0;

void setup()
{
  S1.attach(10);
  S2.attach(9);
  pinMode(Head_S, INPUT);
  pinMode(Comm3, INPUT);
  pinMode(Comm2, OUTPUT);
  pinMode(Comm1, OUTPUT);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(Valve, OUTPUT);

   // Serial.begin(14400);

} //Close void setup

void loop()
{
  digitalWrite (Comm2, LOW);            // Initial mode is FOLDED
  digitalWrite (Comm1, LOW);

//Serial.print("Folde Mode");
//Serial.print("\n");

    //    Serial.print(Force);

  digitalWrite (Valve, LOW);

  digitalWrite (LED1, HIGH);   // Initially all LEDs are ON * * * *
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, HIGH);
  digitalWrite (LED4, HIGH);

  S1.write(115);                // Motors in Folded Position
  S2.write(38);

while (Force > 950)            // Waits until detects the person standing
{
  Force = analogRead (Force_S);  // Read value from Force Sensor
}
Force = 1023;    // Reinitialize for later use

for (i = 1; i <= 5; i += 1) {    // LEDs flash for 5 iterations

  digitalWrite (LED1, LOW);      // - - - -
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, LOW);
  delay (500);

  digitalWrite (LED1, HIGH);      // * - - -
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, LOW);
  delay (100);

  digitalWrite (LED1, HIGH);      // * * - -
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, LOW);
  delay (100);

  digitalWrite (LED1, HIGH);      // * * * -
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, HIGH);
  digitalWrite (LED4, LOW);
  delay (100);

  digitalWrite (LED1, HIGH);      // * * * *
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, HIGH);
  digitalWrite (LED4, HIGH);
  delay (100);

  digitalWrite (LED1, HIGH);      // * * * *
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, HIGH);
  digitalWrite (LED4, HIGH);
  delay (100);

  digitalWrite (LED1, HIGH);      // * * * -
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, HIGH);
  digitalWrite (LED4, LOW);
  delay (100);

  digitalWrite (LED1, HIGH);      // * * - -
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, LOW);
  delay (100);

  digitalWrite (LED1, HIGH);      // * - - -
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, LOW);
  delay (100);

} // Close the for loop

  digitalWrite (LED1, HIGH);      // Lights for Standing Mode * - - *
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, HIGH);

  digitalWrite (Comm2, LOW);            // Standing Mode (data sent to Arduino B)
  digitalWrite (Comm1, HIGH);

for(pos1 = 115, pos2 = 38; pos1 <= 148; pos1 += 1, pos2 -= 1)
{
  S1.write(pos1);
  S2.write(pos2);
  delay(90);
}

delay(7000);      // Waits until S3 and S4 are positioned.

for(pos1 = 148, pos2 = 8; pos1 >= 115; pos1 -= 1, pos2 += 1)
{
  S1.write(pos1);
  S2.write(pos2);
  delay(90);
}

delay(7000);     // Waits for the person to get ready to seat

  digitalWrite (Comm2, HIGH);           // Sitting Mode (data sent to Arduino B)
  digitalWrite (Comm1, LOW);

delay(2000);     // Waits for motors 3 and 4

for(pos1 = 115, pos2 = 47; pos1 >= 60; pos1 -= 1, pos2 += 1)
{
  S1.write(pos1);
  S2.write(pos2);
  delay(90);
}

  digitalWrite (LED1, LOW);      // Lights for Sitting Mode - * * -
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, HIGH);
  digitalWrite (LED4, LOW);

while (Sleep == 0) {              //If head doesn't fall back, keep updating sensor
    Head=digitalRead(Head_S);                   //Read from Sensor
    if (Head==HIGH) {
      delay(5000);              // Check the sensor again after 5 seconds to make sure
         if (Head==HIGH){
           Sleep=1;
         } // Close if 1
         else {
           Sleep=0;
         } // Close else 1
    }   // Close if 2
    else {
      Sleep=0;
    }    // Close else
}  // Close while
Sleep = 0;      // Reinitialize for later use
//////////////////////////////////////////////////////////////////

  digitalWrite (Comm2, HIGH);           // Sleeping Mode (data sent to Arduino B)
  digitalWrite (Comm1, HIGH);

for(pos1 = 60, pos2 = 102; pos1 <= 120; pos1 += 1, pos2 -= 1)
{
  S1.write(pos1);
  S2.write(pos2);
  delay(90);
}

  digitalWrite (LED1, LOW);      // - - - - LEDs OFF
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, LOW);

while (Awake == 0) {              //Wait until the nunchuck detects that the person wakes up
    Wii=digitalRead(Comm3);  //Read from Arduino B
    if (Wii==HIGH) {
      Awake = 1;
    }    // Close if
    else {
      Awake = 0;
    }    // Close else
}  // Close while

Awake = 0;    // Reinitialize for later use

digitalWrite (Valve, HIGH);    // Inflates balloons (Open valve)
delay (800);
digitalWrite (Valve, LOW);     // Close valve

while (Awake == 0) {              //Wait until the nunchuck detects that the person tries to sit
    Wii=digitalRead(Comm3);  //Read from Arduino B
    if (Wii==HIGH) {
      Awake=1;
    }    // Close if
    else {
      Awake=0;
    }    // Close else
}  // Close while

Awake = 0;      // Reinitialize for later use
```

```
  digitalWrite (Comm2, HIGH);           // Sitting Mode (data sent to Arduino B)
  digitalWrite (Comm1, LOW);

  digitalWrite (LED1, LOW);      // Lights for Sitting Mode - * * -
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, HIGH);
  digitalWrite (LED4, LOW);

  delay (4000); // Wait for motors 3 and 4

for(pos1 = 120, pos2 = 42; pos1 >= 70; pos1 -= 1, pos2 += 1)
{
  S1.write(pos1);
  S2.write(pos2);
  delay(90);
}



while (Force > 950)            // Waits until detects the person trying to stand up
{
  Force = analogRead (Force_S);  // Read value from Force Sensor
}
Force = 1023;   // Reinitialize for later use

  digitalWrite (Comm2, LOW);           // Standing Mode (data sent to Arduino B)
  digitalWrite (Comm1, HIGH);

  digitalWrite (LED1, HIGH);      // Lights for Standing Mode * - - *
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, HIGH);

for(pos1 = 70, pos2 = 92; pos1 <= 115; pos1 += 1, pos2 -= 1)
{
  S1.write(pos1);
  S2.write(pos2);
  delay(90);
}

delay(7000);      // Waits for the person to get ready to stand up

  digitalWrite (Comm2, LOW);           // Folded Mode
  digitalWrite (Comm1, LOW);

for(pos1 = 115, pos2 = 47; pos1 <= 148; pos1 += 1, pos2 -= 1)    // Pre-fold
{
  S1.write(pos1);
  S2.write(pos2);
  delay(90);
}
```

```
delay(13000);       // Waits until S3 and S4 are positioned.

for(pos1 = 148, pos2 = 14; pos1 >= 115; pos1 -= 1, pos2 += 1)
{
  S1.write(pos1);
  S2.write(pos2);
  delay(90);
}

for (i = 0; i <= 5; i += 1) {    // LEDs flash for 5 iterations

  digitalWrite (LED1, LOW);      // - - - -
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, LOW);
  delay (500);

  digitalWrite (LED1, HIGH);      // * - - -
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, LOW);
  delay (100);

  digitalWrite (LED1, LOW);       // - * - -
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, LOW);
  delay (100);

  digitalWrite (LED1, LOW);       // - - * -
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, HIGH);
  digitalWrite (LED4, LOW);
  delay (100);

  digitalWrite (LED1, LOW);       // - - - *
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, HIGH);
  delay (100);

  digitalWrite (LED1, LOW);       // - - - *
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, HIGH);
  delay (100);

  digitalWrite (LED1, LOW);       // - - * -
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, HIGH);
  digitalWrite (LED4, LOW);
  delay (100);

  digitalWrite (LED1, LOW);       // - * - -
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, LOW);
  delay (100);

  digitalWrite (LED1, HIGH);      // * - - -
  digitalWrite (LED2, LOW);
  digitalWrite (LED3, LOW);
  digitalWrite (LED4, LOW);
  delay (100);

} // Close the for loop

  digitalWrite (LED1, HIGH);      // Lights for Stand-by Mode * * * *
  digitalWrite (LED2, HIGH);
  digitalWrite (LED3, HIGH);
  digitalWrite (LED4, HIGH);

} // Close void loop
```

```
////////////////////////////////////////////////////////////////////////////////////////////
//                                                                                        //
//          Interactive Inflatable Furniture - Board B (Old Arduino)                      //
//                                                                                        //
//          Henrique Houayek and Ivan Siles                                               //
//                                                                                        //
//      Project #3 - Architectural Robotics (Arch879/ECE893)                              //
//                                                                                        //
//          Spring 2009                                                                   //
//          Clemson University, Clemson, SC                                               //
//                                                                                     //
////////////////////////////////////////////////////////////////////////////////////////////

#include <Servo.h>
#include <Wire.h>
#include "nunchuck_funcs.h"

// Declare all the IN/OUT pin variables

// Inputs
int Comm1 = 11;         // Pin 1 to communicate with Arduino A (Input)
int Comm2 = 2;          // Pin 2 to communicate with Arduino A (Input)

//Outputs
int Comm3 = 3;          // Pin 3 to communicate with Arduino B (Output)

//Servos
Servo S3;
Servo S4;

// Program variables

int Wii = 0;
int pos3 = 0;
int pos4 = 0;
int Arduino_A;
int Mode = 0;

int accy;

void setup()
{
   nunchuck_setpowerpins();
   nunchuck_init();          // send the initilization handshake

   S3.attach(10);
   S4.attach(9);

   pinMode(Comm3, OUTPUT);
   pinMode(Comm2, INPUT);
   pinMode(Comm1, INPUT);
```

```
        Serial.begin(19200);

} // Close void setup

void loop()
{

  digitalWrite (Comm3, LOW);            // Initialize

  S3.write(10);                // Motors in Folded Position
  S4.write(170);

////////////////////////////////////////////////////////////////////////////////

  while (Mode == 0)              // Wait for a response from Arduino A
  {
    Arduino_A = digitalRead (Comm1);   // Read from Arduino A

    if (Arduino_A == HIGH) {
      Mode = 1;      // Mode has changed to standing
    } // Close if
    else {
      Mode = 0;
    } // Close else

  } // Close while
  Mode = 0;    // Reinitialize for later use

                                                      // Standing Mode

for(pos3 = 10; pos3 <= 120; pos3 += 1)
{
  S3.write(pos3);
  delay(90);
} // Close for

for(pos4 = 170; pos4 >= 90; pos4 -= 1)
{
  S4.write(pos4);
  delay(90);
} // Close for

////////////////////////////////////////////////////////////////////////////////

  while (Mode == 0)              // Wait for a response from Arduino A
  {
    Arduino_A = digitalRead (Comm2);   // Read from Arduino A

    if (Arduino_A == HIGH) {
      Mode = 1;      // Mode has changed to sitting
    } // Close if
    else {
      Mode = 0;
    } // Close else

  } // Close while
  Mode = 0;    // Reinitialize for later use

                                                  // Sitting Mode

//for(pos4 = 90; pos4 <= 90; pos4 += 1)
//{
//  S4.write(pos4);
//  delay(90);
//} // Close for

for(pos3 = 120; pos3 <= 180; pos3 += 1)
{
  S3.write(pos3);
  delay(90);
} // Close for



////////////////////////////////////////////////////////////////////////////////

  while (Mode == 0)              // Wait for a response from Arduino A
  {
    Arduino_A = digitalRead (Comm1);   // Read from Arduino A

    if (Arduino_A == HIGH) {
      Mode = 1;      // Mode has changed to sleeping
    } // Close if
    else {
      Mode = 0;
    } // Close else

  } // Close while
  Mode = 0;    // Reinitialize for later use

                                                  // Sleeping Mode

for(pos3 = 180; pos3 >= 150; pos3 -= 1)
{
  S3.write(pos3);
  delay(90);
} // Close for

for(pos4 = 90; pos4 >= 30; pos4 -= 1)
{
  S4.write(pos4);
  delay(90);
} // Close for

////////////////////////////////////////////////////////////////////////////////


while (Wii == 0)      // Check if person wakes up
{
    nunchuck_get_data();
    accy = nunchuck_accely();   // ranges from approx 65 - 173

    Serial.print("Wii position 1: ");
    Serial.print(accy);
    Serial.print("\n");

    if (accy <= 90) {
        Wii=1;
    } // Close if
    else {
        Wii=0;
    } // Close else

} // Close while

Wii = 0;    // Reinitialize for later use

    nunchuck_get_data();
    accy = nunchuck_accely();   // ranges from approx 65 - 173

    Serial.print("Wii position 2: ");
    Serial.print(accy);
    Serial.print("\n");

digitalWrite (Comm3, HIGH);    // Send data to Arduino A (Nunchuck moved)
delay(50);
digitalWrite (Comm3, LOW);
delay(4000);

    nunchuck_get_data();
    accy = nunchuck_accely();   // ranges from approx 65 - 173

    Serial.print("Wii position 3: ");
    Serial.print(accy);
    Serial.print("\n");

while (Wii == 0)      // Check if person wakes up
{
    nunchuck_get_data();
    accy = nunchuck_accely();   // ranges from approx 65 - 173

    Serial.print("Wii position 4: ");
    Serial.print(accy);
    Serial.print("\n");

    if (accy <= 90) {
        Wii=1;
    } // Close if
    else {
        Wii=0;
    } // Close else

} // Close while

digitalWrite (Comm3, HIGH);    // Send data to Arduino A (Nunchuck moved)

////////////////////////////////////////////////////////////////////////////////

                                                      // Sitting Mode

for(pos4 = 30; pos4 <= 90; pos4 += 1)
{
  S4.write(pos4);
  delay(90);
} // Close for

for(pos3 = 150; pos3 <= 180; pos3 += 1)
{
  S3.write(pos3);
  delay(90);
} // Close for



////////////////////////////////////////////////////////////////////////////////

  while (Mode == 0)              // Wait for a response from Arduino A
  {
    Arduino_A = digitalRead (Comm1);   // Read from Arduino A

    if (Arduino_A == HIGH) {
      Mode = 1;      // Mode has changed to standing
    } // Close if
    else {
      Mode = 0;
    } // Close else

  } // Close while
  Mode = 0;    // Reinitialize for later use

                                                      // Standing Mode

//for(pos4 = 90; pos4 >= 90; pos4 -= 1)
```

```
//{
//  S4.write(pos4);
//  delay(90);
//} // Close for

for(pos3 = 180; pos3 >= 120; pos3 -= 1)
{
  S3.write(pos3);
  delay(90);
} // Close for



////////////////////////////////////////////////////////////////////////////////

  while (Mode == 0)              // Wait for a response from Arduino A
  {
    Arduino_A = digitalRead (Comm1);   // Read from Arduino A

    if (Arduino_A == LOW) {
      Mode = 1;      // Mode has changed to folded
    } // Close if
    else {
      Mode = 0;
    } // Close else

  } // Close while
  Mode = 0;    // Reinitialize for later use

                                                      // Folded Mode



for(pos4 = 90; pos4 <= 170; pos4 += 1)
{
  S4.write(pos4);
  delay(90);
} // Close for

for(pos3 = 120; pos3 >= 10; pos3 -= 1)
{
  S3.write(pos3);
  delay(90);
} // Close for

    nunchuck_get_data();
    accy = nunchuck_accely();   // ranges from approx 65 - 173
    delay(4000);

    nunchuck_get_data();
    accy = nunchuck_accely();   // ranges from approx 65 - 173
    delay(4000);


} // Close void loop
```