ARCH 879/ECE 893 - Architectural Robotics
Project 1 - Creative Stimulus for a Child
Team members - Henrique Houayek and Paul Yanik

Interactive Flower to Cultivate Children's Activity

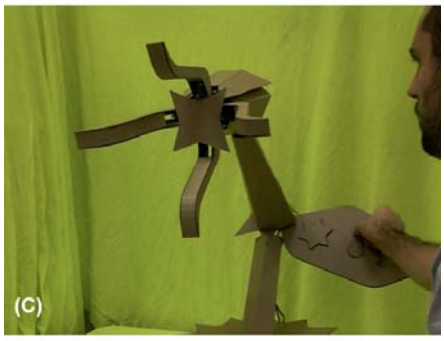
Abstract:

This project presents an interactive flower which aims to cultivate children's creativity by affording an hands on interactive experience which teaches basic notions of flower natural cycle. Initially the children is attracted to the shape, color and lights of the flower. He or she gets puzzle pieces which are forms of three ingredients needed to make the flower bloom: seeds, sun, and rain. As each ingredient is added, an optical sensor detects its placement. Each placement causes the flower to expand by a fraction of its total open configuration. When all three pieces have been place, the flower will have fully bloomed. As each piece is removed, the flower will close by the same fraction. When all pieces have are removed the flower be in its wilted configuration, awaiting another user.

Flower Scenario

(A) Flower is in rest mode

(B) First piece of the puzzle is set, the sun, its petals gently open



(C) Second piece of the puzzle is set, the water, its petals open a bit more

(D) Third piece of the puzzle is set, the seeds, completing the cycle and opening the flower.

(E) Flower in its complete open state

Hardware Used:

- Arduino Duemilanove board
- Adafruit Motor/Stepper/Servo Shield for Arduino
- Futaba servo motor
- Photo resistors (1.5 kΩ – 50 kΩ) (3) (See Figure 1)
- Pullup resistors (6.8 kΩ) (3) (See Figure 1)
- Wooden base
- Articulated arm lamp (as flower stem)
- Lego Mindstorms components (for flower mechanism and motor mount)
- Chipboard cutouts for outer coverings and decoration

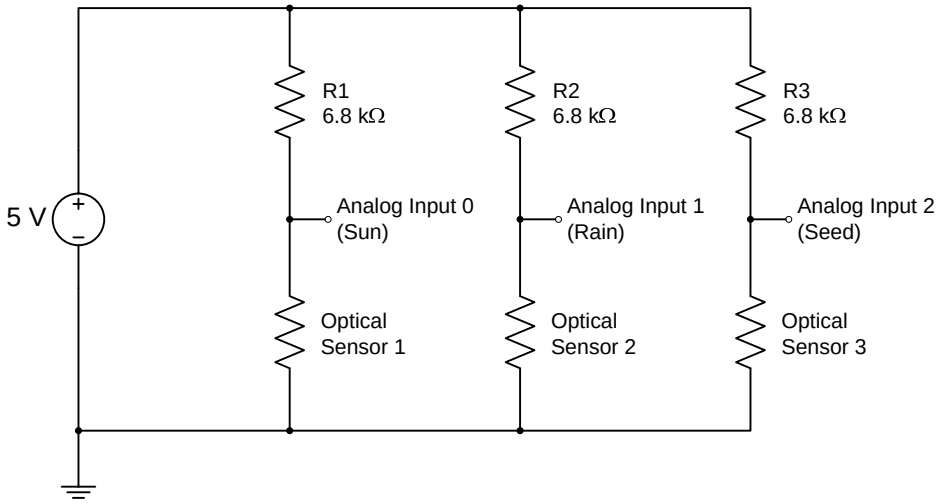


Figure 1 – Sensor schematic for puzzle piece detection.

Source Code:

```
// ------------------------------------------------------
// Filename:      ECE983_Proj1.pde
// Course:        ECE893 - Architectural Robotics
// Team members:  Henrique Houayek, Paul Yanik
//
// Description:
// This design activates a model flower to open (bloom)
// and close based on interactions with a child operator.
// As the child completes a 3-step puzzle, the flower
// opens in stages until it has fully "bloomed."  As
// pieces are removed from the flower closes in stages;
// reversing the flower's progress back to a closed
// configuration.
// ------------------------------------------------------
#include <ServoTimer1.h>
ServoTimer1 servo1;

// ------------------------------------------------------
// Variable Declarations
// ------------------------------------------------------
int inPins[3];              // Array containing the input pin numbers.
int sun = 0;                // Array positions for each pin number.
int rain = 1;
int seed = 2;

int count = 0;              // Variable containing the number of inputs which are
                            // HIGH - indicating a placed puzzle piece.

int last_count = 0;         // Variable containing the number of inputs which were
                            // HIGH on the last interation of the operation loop.

int val = 0;                // Variable containing the value of each input (HIGH or LOW)
                            // as each input is read.

int ANGLE = 50;             // The angular change in position for the servo motor
                            // as each puzzle piece is placed.

// ------------------------------------------------------
// IO Pin Setup
// ------------------------------------------------------
void setup()
{
  servo1.attach(10);

  // Assign pin numbers
  inPins[sun] = 14;
  inPins[rain] = 15;
  inPins[seed] = 16;

  // Set up pins as input or output
  for (int j=0; j<=2; j++) {
    pinMode(inPins[j], INPUT);
  }

  // Open a serial link for onscreen variable monitoring.
  Serial.begin(9600);
}

// ------------------------------------------------------
// Loop for Runtime Operation
// ------------------------------------------------------
void loop()
{
  // Read all inputs and count how many are HIGH
  count = 0;
  for (int q=0; q<=2; q++) {
    val = digitalRead(inPins[q]);
```

```
    if (val == HIGH) {
      count += 1;
    }
  }
  // Display the number of HIGH inputs on the screen
  Serial.println(count);

  // Set the servo angle based on the number of placed
  // puzzle pieces as determined by HIGH inputs

  if (last_count < count) // Indicates that new pieces have been placed.
  {
    // Open the flower to a total angle equal to 'angle' degrees
    // for every placed puzzle piece.
    for (int r=(last_count*ANGLE); r<=(count*ANGLE); r++)
    {
      servo1.write(r);
      delay(40);
    }
  } else
    {
      if (last_count > count) // Indicates that pieces have been removed.
      {
        // Close the flower to a total angle equal to 'angle' degrees
        // for every placed puzzle piece.
        for (int r=(last_count*ANGLE); r>=(count*ANGLE); r--)
        {
          servo1.write(r);
          delay(40);
        }
      }
    }

// Store the number of HIGH inputs for comparison in the next
// loop iteration.
last_count = count;

}
```