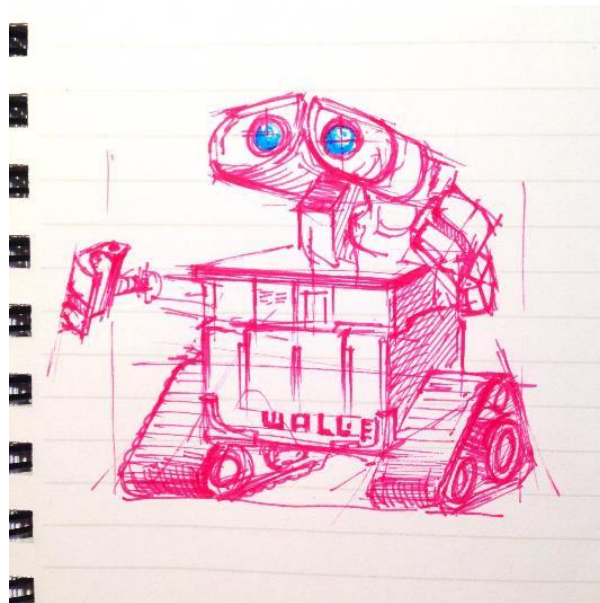


Wall-E Interactive Robot

A reprogrammed WALL-E toy that responds to movements.

By: Maria Kudriavtceva, Paul Golebiewski
& Marryam Khan



Wall-E Interactive Robot Tutorial:

Tools:

Triangle screw driver

Standard star screw driver

Scissors

Arduino

Breadboard

2 RGB lights

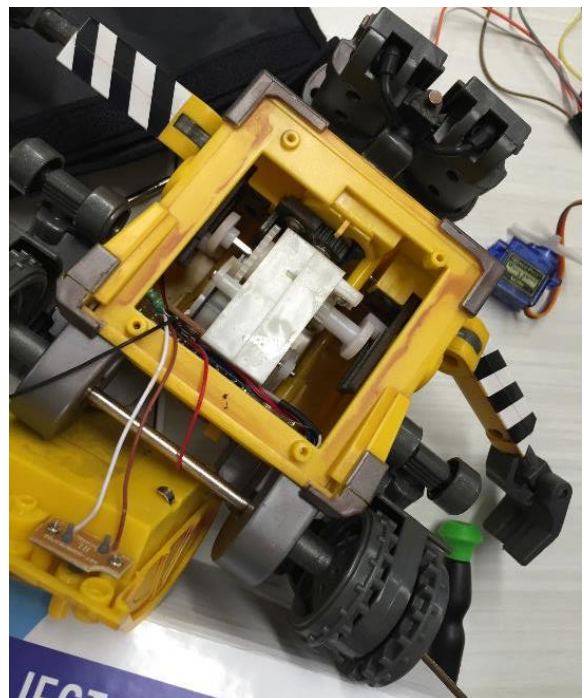
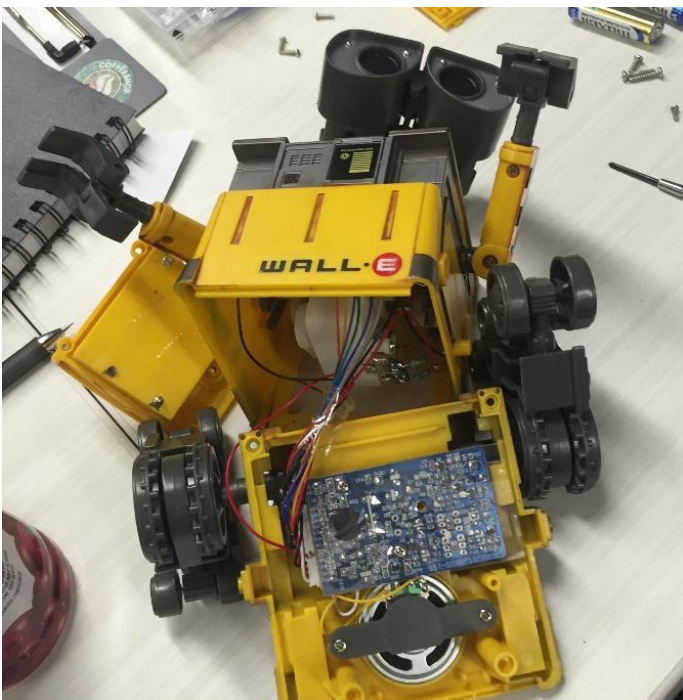
Power Drill

Assorted Sander Attachments

Mini Hand Saw

Toy Deconstruction

Start by opening the toy structure by removing the screws with a triangle screwdriver and standard screwdriver.



Remove all existing toy components from inside the body and disconnect all wires with scissors.

Programming

Eyes:

Start with programming the toy's eyes. Read comments in the code below as well as side notes for detailed instructions.

```
//Start by connecting your RGB lights to the breadboard  
//and the corresponding pins to the Arduino
```

```
//////////LIGHTS//////////  
int redPin1 = 6; //red cable  
int greenPin1 = 10; //blue cable  
int bluePin1 = 7; // yellow cable  
int redPin2 = 11; //white cable  
int greenPin2 = 9; //orange cable  
int bluePin2 = 8; // blue cable  
//////////END//////////
```

Setting up pins for the eyes.

```
void setup()  
{  
  Serial.begin (9600);  
  pinMode(trigPin, OUTPUT);  
  pinMode(echoPin, INPUT);  
  pinMode(redPin1, OUTPUT);  
  pinMode(greenPin1, OUTPUT);  
  pinMode(bluePin1, OUTPUT);  
  pinMode(redPin2, OUTPUT);  
  pinMode(greenPin2, OUTPUT);  
  pinMode(bluePin2, OUTPUT);  
  
  myservo1.attach(servo1Pin);  
  myservo2.attach(servo2Pin);  
  
  myservo3.attach(servo3Pin);  
  //myservo4.attach(servo4Pin);  
}  
  
//Here we create the function for changing the eye color  
void loop()  
{  
  eyeMood();  
}
```

For now disregard the code pertaining to the micro-servo motors.

```

//Turn the proper color ON
void setColor(int red, int green, int blue)
{
    analogWrite(redPin1, red);
    analogWrite(greenPin1, green);
    analogWrite(bluePin1, blue);
    analogWrite(redPin2, red);
    analogWrite(greenPin2, green);
    analogWrite(bluePin2, blue);
}

```

To set the different eye colors use this function and RGB values.

```

//////////////////EYES//////////////////

```

```

// BLUE eyes
void blueEyes()
{
    setColor(0, 0, 255); //blue
}

```

```

// GREEN eyes
void greenEyes()
{
    setColor(0, 255, 0); //green
}

```

```

// RED eyes
void redEyes()
{
    setColor(255, 0, 0); //red
}

```

```

//Turn eyes OFF
void eyesOff()
{
    setColor(0, 0, 0); //off
}

```

```

//////////////////GENERAL//////////////////

```

```

//Change eye color depending on the distance to closest obstacle

```

```

void eyeMood()
{
    long duration, distance;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW );
    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2) / 29.1;

    if (distance <= 10)
    {
        → redEyes();
        handsUP();
        headDOWN();
    }
}

```

```

    else if (distance > 10 && distance <= 15)
    {
        → greenEyes();
        handsDOWN();
        headUP();
    }

```

```

    else if (distance > 15 && distance <= 30)
    {
        → blueEyes();
    }

```

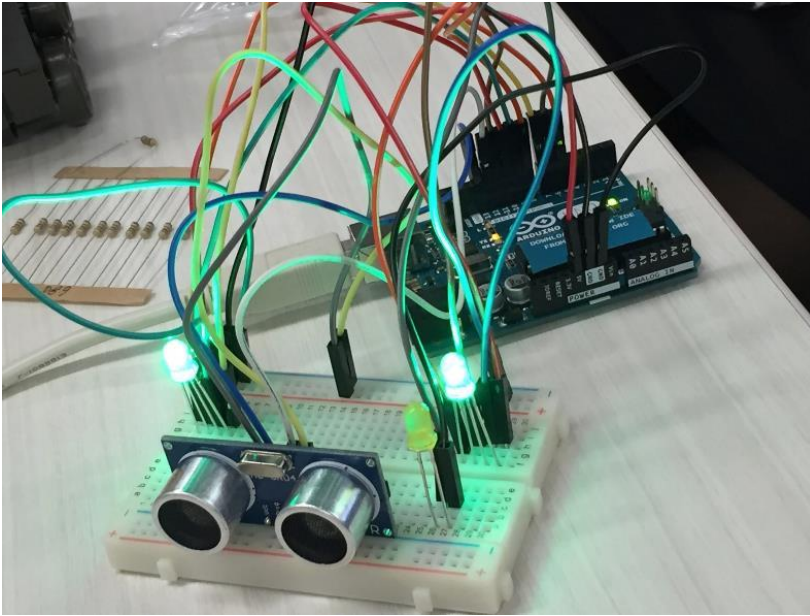
```

    else if (distance >= 30)
    {
        → eyesOff();
        handsDOWN();
        headDOWN();
    }
    delay(200);
}

```

Use this function with an ‘if statement’ to change the color of the RGB lights in relation to proximity. When the distance of an object from the WALL-E is greater than 15 cm, the RGB lights remain blue, in their “natural” state.

Likewise, if the distance of the object is between 10 and 15 cm the RGB lights will turn from blue to green and if less than 10cm then green to red.



Once your RGB lights are functioning connect the Ultra-sonic sensor to the breadboard and Arduino with wires.

Arms:

For the arms we attached two micro servo motors to the breadboard and Arduino with wires. Here we are setting up the pins.

```
/////////////////ARMS/////////////////  
int servo1Pin = 5;  
int servo2Pin = 4;  
int handPos = 0; // Set initial hads position  
Servo myservo1; // Create RIGHT HAND  
Servo myservo2; // Create LEFT HAND  
/////////////////END/////////////////
```

Add micro-servo 1 and 2 to the void setup function like so.

```
myservo1.attach(servo1Pin);  
myservo2.attach(servo2Pin);
```

```

//////////ARMS//////////

//Put the hands up
void handsUP()
{
    for(handPos; handPos <= 90; handPos++) // hands moving UP
    {
        //Put right hand UP
        myservo1.write(handPos);
        //Reverse motion for left hand to go up as well
        myservo2.write(90-handPos);
        //Set motion speed
        delay(10);
    }
}

//Put the hands down
void handsDOWN()
{
    for(handPos; handPos >= 0; handPos--)
    {
        myservo1.write(handPos);
        myservo2.write(90-handPos);
        delay(10);
    }
}

```

Here we are programming the arms (as hands) to move up and down from 0 to 90 degrees with a continuous 'for loop'.

```

//////////GENERAL//////////

//Change eye color depending on the distance to closest obstacle
void eyeMood()
{
    long duration, distance;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2) / 29.1;

    if (distance <= 10)
    {
        redEyes();
        handsUP();
        headDOWN();
    }

    else if (distance > 10 && distance <= 15)
    {
        greenEyes();
        handsDOWN();
        headUP();
    }

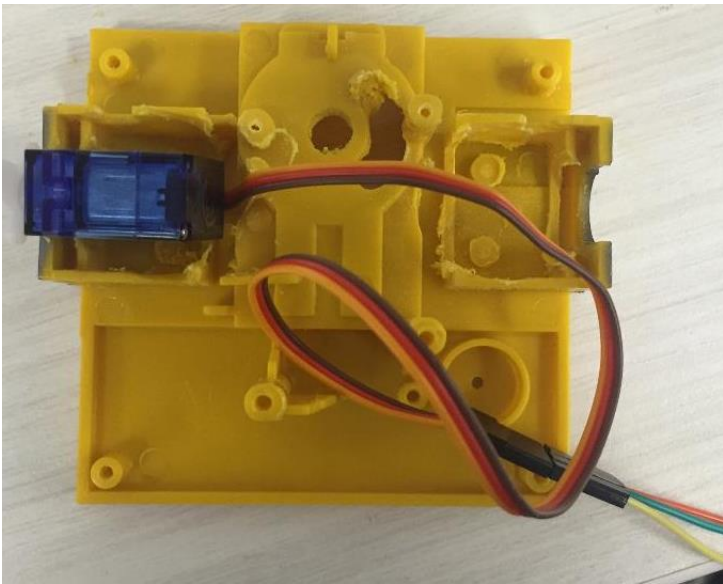
    else if (distance > 15 && distance <= 30)
    {
        blueEyes();
    }

    else if (distance >= 30)
    {
        eyesOff();
        handsDOWN();
        headDOWN();
    }

    delay(200);
}

```

In the code above, we have programmed the arms (as hands) to move 90 degrees in correspondence with the distance that the proximity sensor is detecting as we did for the eyes.



Micro-servo motor for arms inside WALL-E structure

Head:

```

//////////////////HEAD//////////////////
int servo3Pin = 2;
int headPos = 0; // Set initial head position
Servo myservo3; // Create HEAD
//////////////////END//////////////////

```

For the head we attached a micro servo motor to the breadboard and Arduino in the same manner. Then we set up the pins just as before.

```

//////////////////HEAD//////////////////
void headUP()
{
    for(headPos; headPos <= 50; headPos++) // head moving UP
    {
        myservo3.write(headPos);
        delay(20);
    }
}

void headDOWN()
{
    for(headPos; headPos >= 0; headPos--)
    {
        myservo3.write(headPos);
        delay(20);
    }
}

```

The toy's head is programmed similarly with a 'for loop' to change its position 50 degrees as the proximity of the object changes.

```

/////////////////GENERAL/////////////////

//Change eye color depending on the distance to closest obstacle
void eyeMood()
{
    long duration, distance;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW );
    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2) / 29.1;

    if (distance <= 10)
    {
        redEyes();
        handsUP();
        → headDOWN();
    }

    else if (distance > 10 && distance <= 15)
    {
        greenEyes();
        handsDOWN();
        → headUP();
    }

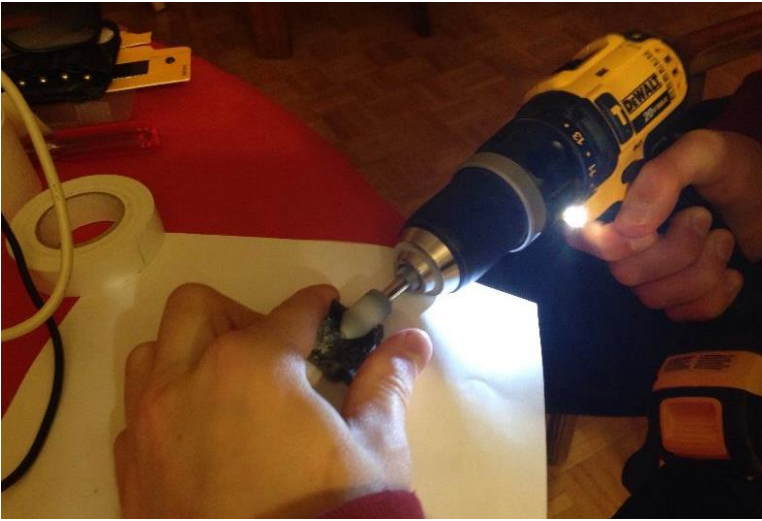
    else if (distance > 15 && distance <= 30)
    {
        blueEyes();
    }
    else if (distance >= 30)
    {
        eyesOff();
        handsDOWN();
        → headDOWN();
    }
    delay(200);
}

```

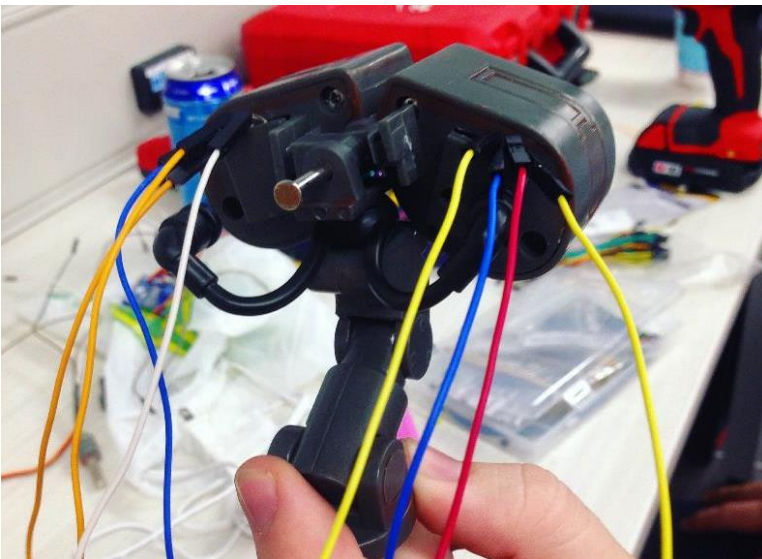
In the code above we are calling the head function as we did with the eyes and arms.

Assembly Process:

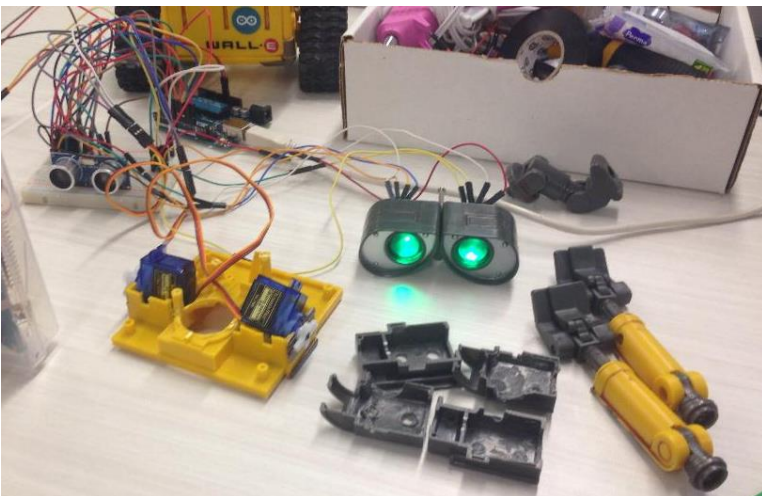
To prepare the structure to encase our breadboard, wires, micro-servo motors and other components we reshaped it using a sander attachment on a power drill



Reshaping WALL-E's head



The assembled head component with wiring



Some reshaped structural components and functioning electronic components

Good Luck!

