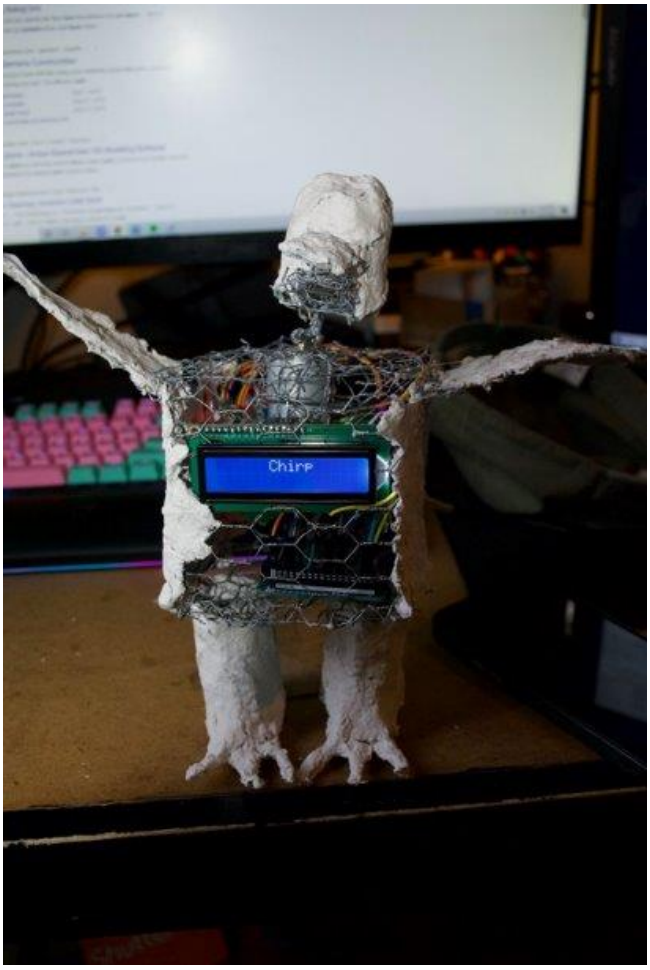
Bird Puppet

By Connor Kelley, Kaine Osborn, Lu Haibo, Jeffrey Doman

Overview:

This is an instructable guide to building an animatronic bird puppet with multiple moving parts that can be displayed on a table, desk, or another flat surface. It is important that all steps be followed exactly, and the code is set up properly to achieve the desired result.



Supplies

Arduino:

ATmega 2560

20 Male-Female Wires

Speaker:

uxcell 0.25W speaker

(a15071300ux0165)

LCD:

Sunfounder 1602 LCD

Trimmer 10k Breadboard Potentiometer

Equipment:

- Chicken Wire
- Pliers
- Ruler
- Paper Mache Mix
- Water
- Newspaper or Towel
- Bowl or Pot
- Tape

Step 1: Stepper Motor System Setup

1. Open Atmel Studio, click create a new project, then enter a name of your choice and save it in a destination you will remember. Click next and select the Atmega 2560 device. Enter the following code

```c
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include "Debugger.h"
#include "agitate.h"
#include "LCD.h"
void fill_wash();
void io_init(void);
int main(void)
{
    io_init();
    Init_PORTs();
    Init_LCD();


    char T_mess[] = {"Chirp"};
    char B_mess1[] = {"Test"};
    char B_mess2[] = {"Test"};

        LCD_write(INSTR_WR, 0x01);
        _delay_ms(2);   //can remove if use Busy Flag check

        LCD_write(INSTR_WR, 0x02);
        _delay_ms(2);  //can remove if use Busy Flag check
        //
        LCD_write(INSTR_WR, 0x86);
        Print_string (T_mess);


        LCD_write(INSTR_WR, 0xC0);
        _delay_us(50);  //can remove if use Busy Flag check
    while (1)
    {
        Stepper_drive(12); //STEPPER DRIVE FUNCTION FOR 12 SECONDS
        LCD_write(INSTR_WR, 0xC6);  // Set DDRAM address to bottom row, left character
        _delay_us(50);  //can remove if use Busy Flag check
        Print_string (T_mess);
        _delay_ms(2000);
        LCD_write(INSTR_WR, 0xC6);  // Set DDRAM address to bottom row, left character
        _delay_us(50);  //can remove if use Busy Flag check
        Print_string (T_mess);
        _delay_ms(2000);

    }
}

void io_init(void)
{
    DDRL = 0xFF; //MOTOR
    PORTL= 0x00;
    PORTA = 0xFF;
    DDRA = 0x00; //PUSHBUTTON AND SWITCHES
    DDRC =0xFF; //LED's
    PORTC = 0x00;
    DDRB = (1<<PB0);
    PORTB = (1<<PB0);


}
```

2. Once Main.c is entered, the .h and .c Files will need to be created. In the “Solution Explorer” tab right click on the name project, click add, and then new item. Select .c File and name it whatever you would like.
3. Enter the following code into the .c

```c
#include "agitate.h"

uint8_t Spin[4] = {0x01, 0x02, 0x04, 0x08};
uint8_t Agitate[4] = {0x03, 0x06, 0x0C, 0x09};
uint8_t Agitatereverse[4] = {0x09, 0x0C, 0x06, 0x03};
uint8_t x =1;

void Stepper_drive(uint8_t rev)
{
    uint16_t steps; //STEPS INITIALIZED
    for(uint16_t y =0; y<(rev/4); y++)
    {
        steps = 1000;
        for(uint16_t i =0; i<steps; i++)
        {
            for(uint16_t j=0; j<4; j++)
            {
                Stepper_output = Agitate[j]; //AGITATE MOTOR
                Stepper_output2 = Agitate[j]; //AGITATE MOTOR


                _delay_ms(3); //DELAY 3 MS
            }
        }
        for(uint16_t i=0; i<steps; i++)
        {
            for(uint16_t j=0; j<4; j++)
            {
                Stepper_output = Agitatereverse[j]; //AGITATE IN REVERSE DIRECTION
                Stepper_output2 = Agitatereverse[j]; //AGITATE IN REVERSE DIRECTION
                //Stepper_output3 = Agitatereverse[j]; //AGITATE IN REVERSE DIRECTION
                _delay_ms(3); // DELAY 3MS
            }
        }
    }
}
void Stepper_drive2(uint8_t rev)
{
    uint16_t steps; //STEPS INITIALIZED
    for(uint16_t y =0; y<(rev/4); y++)
    {
        steps = 400/2;
        for(uint16_t i =0; i<steps; i++)
        {
            for(uint16_t j=0; j<4; j++)
            {
                Stepper_output2 = Agitate[j]; //AGITATE MOTOR
                _delay_ms(3); //DELAY 3 MS
            }
        }
        for(uint16_t i=0; i<steps; i++)
        {
            for(uint16_t j=0; j<4; j++)
            {
                Stepper_output2 = Agitatereverse[j]; //AGITATE IN REVERSE DIRECTION
                _delay_ms(3); // DELAY 3MS
            }
        }
    }
}
```

```
void Stepper_drive3(uint8_t rev)
{
    uint16_t steps; //STEPS INITIALIZED
    for(uint16_t y =0; y<(rev/4); y++)
    {
        steps = 75;
        for(uint16_t i =0; i<steps; i++)
        {
            for(uint16_t j=0; j<4; j++)
            {
                /*Stepper_output3 = Agitate[j]; //AGITATE MOTOR*/
                _delay_ms(3); //DELAY 3 MS
            }
        }
        for(uint16_t i=0; i<steps; i++)
        {
            for(uint16_t j=0; j<4; j++)
            {
                /*Stepper_output3 = Agitatereverse[j]; //AGITATE IN REVERSE DIRECTION*/
                _delay_ms(3); // DELAY 3MS
            }
        }
    }

}

void motor_spin(void)
{
    uint16_t steps; //STEPS INITIALIZED
    for(uint8_t h=0; h<x; h++)
    {
        steps = 3000/4; //STEPS SET
        for(uint16_t i =0; i<steps; i++)
        {
            for(uint16_t j=0; j<4; j++)
            {
                Stepper_output = Spin[j]; //SPIN WASH
                _delay_ms(3); //DELAY 3 MS
            }
        }
    }
    PORTC=0x08; //DONE LED
}
```

4. Repeat the following action for the .h and enter the following code into your new .h file.

```
#ifndef AGITATE_H_
#define AGITATE_H_
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

void motor_spin(void);
void Stepper_drive(uint8_t spin);
void Stepper_drive2(uint8_t rev);
void Stepper_drive3(uint8_t rev);
void Stepper_init(void);

#define Stepper_output (PORTA)
#define Stepper_output2 (PORTC)
/*#define Stepper_output3 (PORTA)*/


extern uint8_t Spin[4];
extern uint8_t Agitate[4];
extern uint8_t Agitatereverse[4];




#endif /* AGITATE_H_ */
```

5. After all the code is written, go to the “File” tab and save the project
6. Follow the Below Pinout to connect the stepper motor to your Micro Controller

For all 3 Motors connect the + to 5V and the – to ground

Wing Stepper Motor:

MOTOR IN1: PL0

MOTOR IN2: PL1

MOTOR IN3: PL2

MOTOR IN4: PL3

Head/Beak Stepper Motor:

MOTOR IN1: PA0

MOTOR IN2: PA1

MOTOR IN3: PA2

MOTOR IN4: PA3

Step 2: LCD System Setup

1. The Main.c program from the mechanical system will already have the USART code. The only thing needed is the .c and .h files shown in the pictures below. In the "Solution Explorer" tab right click on the name project, click add, and then new item. Select .c File and name it whatever you would like.

```c
#include <avr/io.h>
#include "LCD.h"

void Init_PORTs (void)
{
    DDRL = 0xFF;    /* PORTL all outputs */
    PORTL = 0x00;   /* Set all outputs to 0 initially */
    DDRD = 0x07;    /* Set D.0 through D.2 to outputs for LCD control */

}

void Init_LCD (void)
{

    _delay_ms(35);              /* wait for more than 30mS after VDD rises to 4.5V */
    LCD_write(INSTR_WR,0x38);   /* function set 8bits, 2line, display off */
    _delay_us(50);              /* wait for more than 39microS */
    LCD_write(INSTR_WR,0x0C);   /* display on, cursor off, blink off */
    _delay_us(50);              /* wait for more than 39microS */
    LCD_write(INSTR_WR,0x01);   /* display clear */
    _delay_ms(2);               /* wait for more than 1.53mS */
    LCD_write(INSTR_WR,0x06);   /* entry mode set, increment mode */
}
/* End of LCD initialization */

void LCD_write (unsigned char RS, unsigned char data)
{
    if(RS==DATA_WR) PORTD = 0b00000001; /* write data: E = 0 R/!W=0, (write)RS = 1 ,   */
    else            PORTD = 0b00000000; /* Write instruction: RS = 0 E = 0, R/!W=0 (write) */


    PORTD = PORTD | 0x04;   /* Take E HIGH (logic 1) */
    PORTL = data;
    _delay_us(50);          /* needs to be at least 30uS or no display - use 50 */
    PORTD = PORTD & 0x01;   /* Take E LOW (logic 0) */
    _delay_us(50);          /* Delay REQUIRED */
}

void Print_string(char *str_ptr)
{
    PORTD = 0b00000001; /* write data: RS = 1 E = 0, R/!W=0 (write) */

    while(*str_ptr != '\0')
    {
        PORTD = PORTD | 0x04;   /* Take E HIGH (logic 1) */
        PORTL = *str_ptr++;
        _delay_us(50);          /* needs to be at least 30uS or no display - use 50 */
        PORTD = PORTD & 0x01;   /* Take E LOW (logic 0) */
        _delay_us(50);          /* Delay REQUIRED */
    }
}
```

2. Repeat the following action for the .h and enter the following code
3. into your new .h file.

```c
#ifndef LCD_H_
#define LCD_H_
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

#define INSTR_WR    0
#define DATA_WR     1

void Init_PORTs (void);
void Init_LCD (void);
void LCD_write (unsigned char RS, unsigned char data);
void Print_string(char *str_ptr);



#endif /* LCD_H_ */
```

4. After all the code is written, go to the “File” tab and save the project
5. Follow the Below Pinout to connect the LCD to your Micro Controller

VSS : Ground

VDD: 5V

VE: Middle Pin of Potentiometer (One of the outer pins of potentiometer goes to ground other goes to 5V)

RS: Pin D0

RW: Pin D1

EN: Pin D2

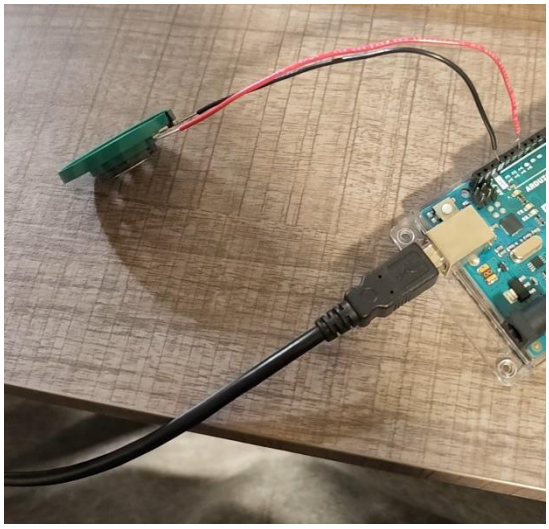
D:0-D:7 : PORTL 0-7

A: 5V

K: Ground

Optional Step 3: Coding the Speaker

1. Since the project target is to make a bird-like sound, we will be utilizing Pulse Width Modulation feature from Arduino 2560 to produce a 7800 HZ square wave to drive the speaker.



2. Connect Speaker gnd(Ground) [Black Wire] to Arduino 2560's gnd pin. Connect Speaker 5v (Signal)[Red Wire] to Arduino 2560's 11 pin (OCR1A). This is a simple connection.

```c
/*
 * speaker.c
 * Created: 2022/11/28 15:37:24
 * Author : Haibo Lu
 */
#include <avr/io.h>
#define F_CPU 16000000UL // F_CPU must be defined BEFORE the #include <util/delay.h>
#define  BAUD 9600
#include "util/delay.h"

int main(void)
{

    DDRB =  0b11111111;  //PB set as output          Port B is  pwm
    PORTB = 0b00000000;  // turn it off at first

    //OCR1A =   511 ;           //duty cycle
    TCCR1A = (1<<COM1A1) | (1<<COM1A0) | (1<<WGM10);    //inverting mode    9bit timer   0b11000010
    TCCR1B = (1<<WGM12)| (1<<CS11);           //prescaler is 8     1 7800Hz pwm        0b00001010

    while (1)
    {
        OCR1A = 50  ;           //duty cycle
        _delay_ms(300);
        OCR1A = 400 ;           //duty cycle
        _delay_ms(100);

        OCR1A = 50  ;           //duty cycle
        _delay_ms(300);
        OCR1A = 400 ;           //duty cycle
        _delay_ms(100);

        OCR1A = 50  ;           //duty cycle
        _delay_ms(300);
        OCR1A = 400 ;           //duty cycle
        _delay_ms(100);

        OCR1A = 511 ;           //duty cycle
        _delay_ms(2000);
    }
}
```

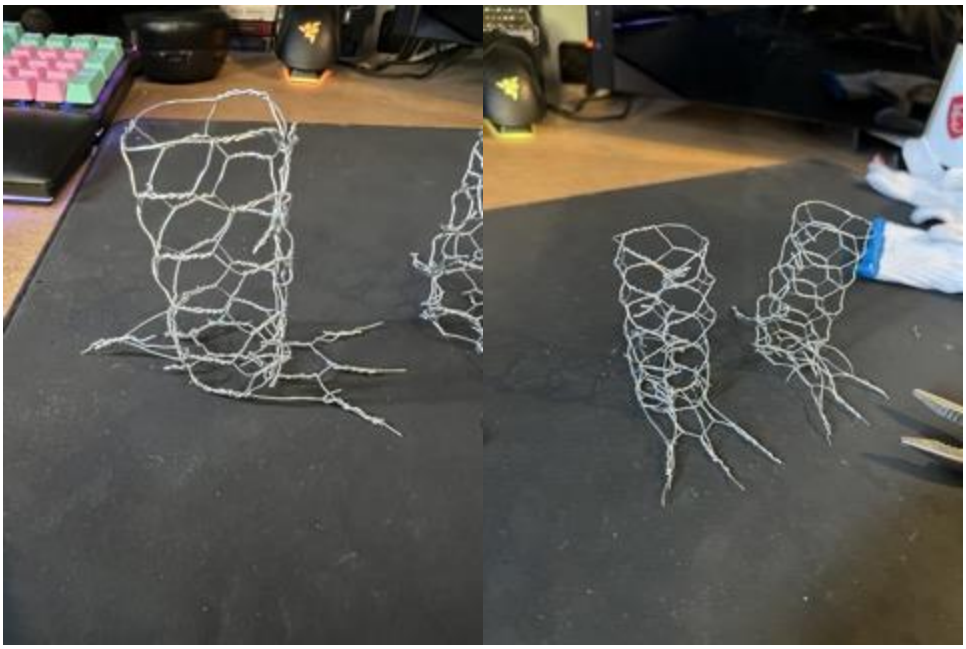
3. Add C code from above picture to your Main.c file. It uses inverting mode and 9-bit counting flag.

4. build project and upload .hex file to Arduino 2560. You should hear a vivid high frequency bird sound.

Step 4: Construction of the Birds Body

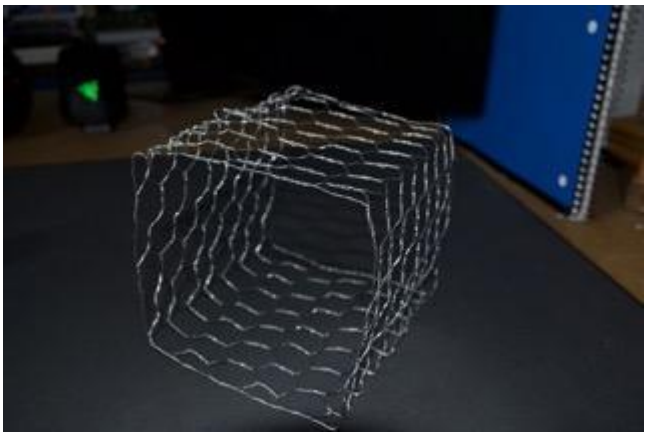
1. Building the Legs

To build the legs, the first step is to make a circular leg piece. I cut out a 3in by 4.5in piece. Then wrap the piece around something circular, I used my fingers and interlocked the ends. The resulting cylinder should be 1.5in diameter and 3in tall. I used a 2in piece of extra wire to go through both sides and twisted the wire to connect the pieces better. We will use this method to connect wires a lot. I used the extra wire at the top, bottom, and middle to make sure it could be secure. Then we needed to make the feet. Cut out a 2in by 3in piece. Cut one of the short ends into 3rds and pull it apart to make 3 claws. To attach the feet to the leg we will use the extra wire method mentioned earlier. You can use as many connections as needed to secure it in place. Then to give the feet some volume, I added a wire connected from the tip of the feet to about half an inch up on the leg. I did the same for the back of the foot, with only one in the middle. Repeat the process for the other leg. The result is shown below.

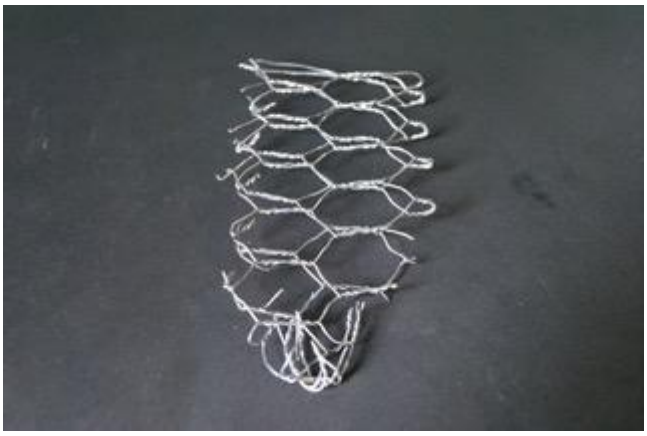


2. Building the body

First, cut out a 4in by 16.5in piece of wire frame. Then every 4in fold the piece at a 90-degree angle to make a box. To connect the box using the wire twisting method. It should look like the image below.

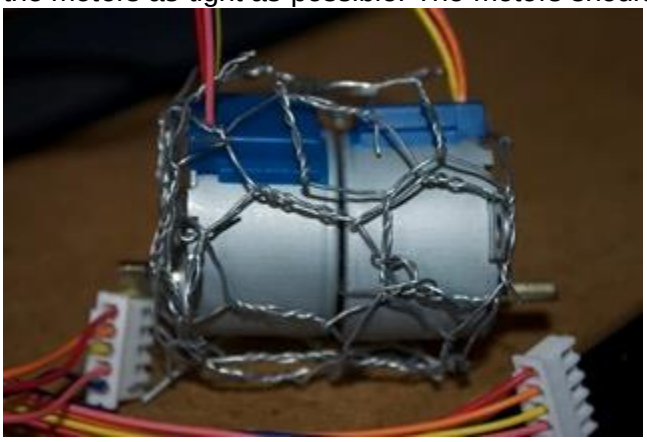


Then cut out 2 4in by 4.5in pieces and attach one to one of the sides, this will be the bottom, we will save the other one for the top. The next step is to make the wings. Cut out a 4in by 5in piece. Fold the piece in half, and attach the open ends together. Use the wire cutters to shape the bottom of the wing, I cut the front at a 45-degree angle for about half an inch, and the back at a 60-degree angle for one inch up the side. You may need to add extra wires to make sure they are still connected. The result is shown below. Repeat this process to make the other wing.



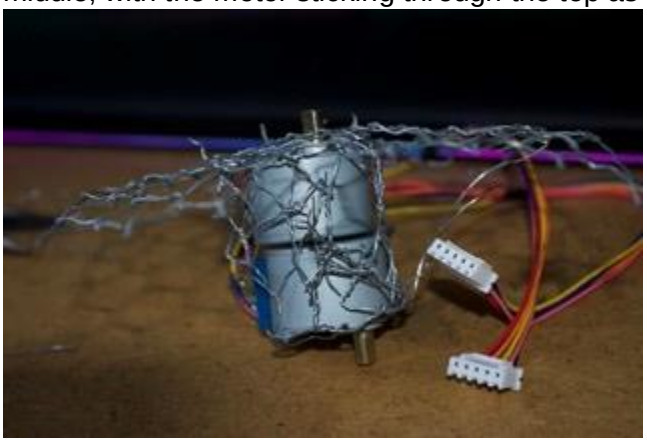
To connect the wings to the body, use the extra wire method. I connected them on the last ring below the top of the body. Do not make the connection too tight, as we want the wings to be able to rotate up and down.

We need to make a cage for the motors to sit in. To do this take a 5x5 piece and wrap it around the motors as tight as possible. The motors should be placed back to back as shown below.



Motor Cage

Then attach this motor cage to the top part we cut out earlier and saved. It should go right in the middle, with the motor sticking through the top as shown below.



3. Building the head

To make the head, cut out a 4in by 4in piece. Then take something cylindrical, I used a medicine bottle, and push the piece down on top, wrapping the edges around as shown below.

Add in some extra connections to make sure it will hold its shape and the head frame is halfway done. The next step is to make the two beaks. To make the beak, cut a 2in x 3in piece. Take the piece and wrap it around the tip of your pinky finger to make a cone-shape. Add in some extra wire to secure it in that shape, then press the pointed end flat to make it into a triangle. Attach one piece halfway up the head and secure it so it cannot move. Attach the second piece below it, only connecting the bottom of the piece. This makes it so the piece can rotate up and down. The final result is shown below.

We need a way to make the head connect to the motor. To do this take a piece of wireframe about 2in by 2in. Roll the piece up as shown below.

Connect this piece to the bottom of the frame across the middle as shown.

Step 5: Applying Paper Mache to the Bird

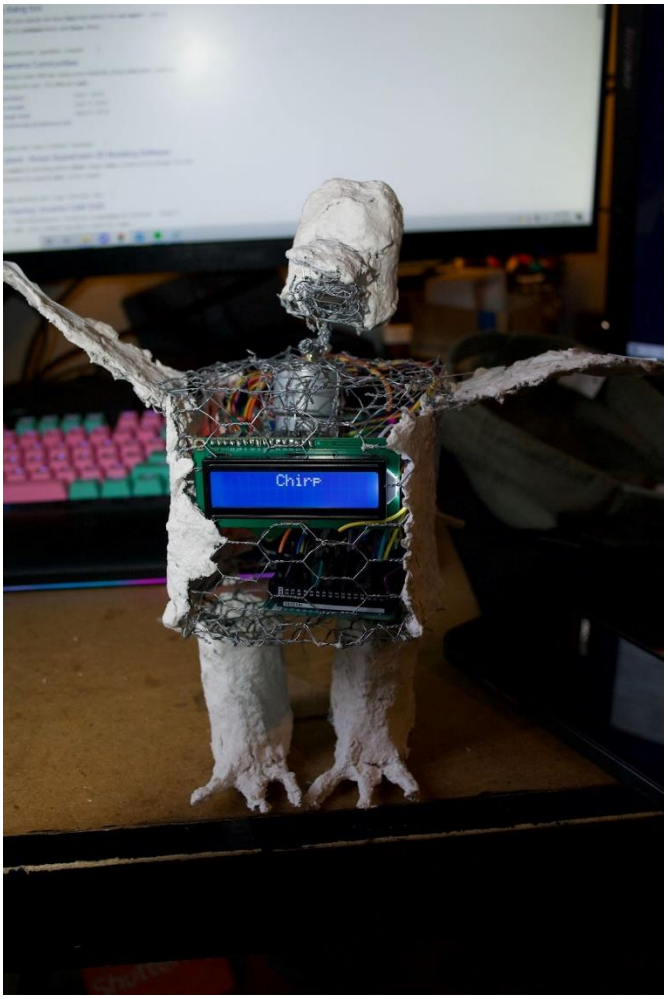
To do the paper mache, get a large pot or bowl to make the mix. Follow the instructions on the packaging to make the mix. Once it feels like dough, start applying it to the wings, head, feet, and only the sides of the body. On the legs, leave a little bit of room at the top so you can attach them to the body later on. You need to be able to access the body so we can add in the rest of the paper mache after those parts are in. It will take about 24 hours to dry.

Step 6: Placement of Parts Inside Bird

The Next step is to take a long piece of string, at least 2 feet as you can always cut it shorter. Wrap the middle around the axle of the bottom motor and put a piece of tape around it. Place the Arduino on the bottom of the board with the port facing backward. Line up the LCD screen along the front and cut out a space for it to sit in. You can add an extra wire around the screen to help secure it. Push all the wires down careful to not pull anything out. Then take the motor and top piece and place it on top with the strings hanging over the top toward the wings. Tie the string to the end of the wings and use tape to help secure it. Then use the extra wire method to secure the top to the body.

Next, we need to connect the head. To do this, take a piece about 4x3in and roll it up as tight as possible. Take the top edge and insert it into the crossing piece on the head. Then using pliers pull apart some of the wire to make a spot to attach to the motor. Make a 90-degree angle between the open part and the head so it sits upright. Place the neck over the motor piece and use pliers to hold it tight to the piece.

Next up attach the legs. Place them under the body to find where they need to go to balance the body. Use the wire method to secure the legs to the bottom of the body.



Plug it in and watch your bird come to life!

Resources

1. Utmel, "28BYJ-48 5V Stepper Motor: 28BYJ-48 datasheet, pinout, Wiring," *Utmel*. [Online]. Available: https://www.utmel.com/components/28byj-48-5v-stepper-motor-28byj-48-datasheet-pinout-wiring?id=1294. [Accessed: 27-Oct-2022].
2. Alldatasheet.com, "Atmega2560 datasheet(pdf) - atmel corporation," *ALLDATASHEET.COM - Electronic Parts Datasheet Search*. [Online]. Available: https://www.alldatasheet.com/datasheet-pdf/pdf/107092/ATMEL/ATMEGA2560.html. [Accessed: 27-Oct-2022].
3. "PowerCore 5000," *Anker*. [Online]. Available: https://www.anker.com/products/a1109?variant=37437604724886. [Accessed: 27-Oct-2022].
4. "Resources." *Cattywampus Puppet Council*. [Online]. Available: https://cattywampuspuppetcouncil.com/resources/. [Accessed: 27-Oct-2022].
5. "Marionnette Puppets - Big Fluffy Bird Craft," *Hanbury House*. [Online]. Available: https://hanburyhouse.com/2011/01/04/fluffy-big-marionnette-bird-puppets/. [Accessed: 27-Oct-2022].
6. "How to Make a Paper Mache Sculpture with Chicken Wire." *EHow Leaf Group*, [Online]. Available: https://www.ehow.com/how_12098299_make-paper-mache-sculpture-chicken-wire.html. [Accessed: 27-Oct-2022].
7. "Bird Puppet Pattern," *The Tucson Puppet Lady*. [Online]. Available: https://thetucsonpuppetlady.com/downloads/bird-pattern/. [Accessed: 27-Oct-2022].
8. "What Is Needed to Move a Puppet?" -*Robot Parts Robotshop Community*. [Online]. Available: https://www.robotshop.com/community/forum/t/what-is-needed-to-move-a-puppet/25344. [Accessed: 27-Oct-2022].
9. *"*(PDF) The Dynamic Modeling of a Bird Robot *"Researchgate*. [Online]. Available: https://www.researchgate.net/publication/242655176_The_dynamic_modeling_of_a_bird_robot. [Accessed: 27-Oct-2022].
10. "28BYJ-48 - 5V Stepper Motor," *Components101*. [Online]. Available: https://components101.com/motors/28byj-48-stepper-motor. [Accessed: 27-Oct-2022].
11. Frame Structure
12. "Bird wooden flying puppet kits," *Baker Ross*. [Online]. Available: https://www.bakerross.com/bird-wooden-flying-puppet-kits-1. [Accessed: 27-Oct-2022].
13. Tamuria, "5 easy puppets to make with paper and cardboard," *Gleeful Grandiva*, 26-May-2022. [Online]. Available: https://gleefulgrandiva.com/2019/03/18/5-easy-puppets-to-make-with-paper-and-cardboard/. [Accessed: 27-Oct-2022].
14. "How to make a paper mache sculpture with Chicken Wire," *eHow*. [Online]. Available: https://www.ehow.com/how_12098299_make-paper-mache-sculpture-chicken-wire.html. [Accessed: 27-Oct-2022].
15. "How to make a paper mache sculpture with Chicken Wire," *eHow*. [Online]. Available: https://www.ehow.com/how_12098299_make-paper-mache-sculpture-chicken-wire.html. [Accessed: 27-Oct-2022].
16. Rosemarybeetle and Instructables, "Puppet making: Painting a Papier Mache puppet head," *Instructables*, 07-Nov-2017. [Online]. Available: https://www.instructables.com/Painting-a-puppet-head/. [Accessed: 27-Oct-2022].
17. ""Amazon.com: Uxcell 2Pcs 8 ohm 0.25W 27mm DIA external magnet mini ..." [Online]. Available: https://www.amazon.com/uxcell-0-25W-External-Magnet-Loudspeaker/dp/B018HMJIZU. [Accessed: 27-Oct-2022].
18. "Arducam 1602 16x2 LCD display module based on HD44780 controller character white on blue with backlight for Arduino," uctronics. [Online]. Available: https://www.uctronics.com/display/arducam-1602-16x2-lcd-display-module-based-on-hd44780-controller-character-white-on-blue-with-backlight-for-arduino.html. [Accessed: 27-Oct-2022].