

HAUNTED DOLLHOUSE REPAIR MANUAL

**EE-490 SENIOR DESIGN PROJECT
B-SECTION FALL 2006**

12/10/2006

CREATED BY:

Michael Hwang
Tyson Green
Jim Heakins
Brandon Koenigsknecht

FOREWORD

This manual contains an introductory description on the Haunted Dollhouse and procedures for its inspection/service and overhaul of its main components.

Other information considered as generally known is not included.

Read this manual to familiarize yourself with the dollhouse and its maintenance. Use this section as well as other sections to use as a guide for proper inspection and service. This manual will help you know the dollhouse better so that you can provide fast and reliable service/maintenance.

- This manual has been prepared based on the latest specifications at the time of publication.
- Illustrations in the manual show the basic principles of construction/operation.

ACKNOWLEDGEMENTS

The authors of this repair manual express sincere appreciation to everyone who assisted in any manner in making our project a success, with special thanks to the following:

For parts and design support:

Gerald Kozlowski, Sr. Lab Coordinator Electrical & Computer Engineering,
Kettering University

Dr. Bhag Guru, Professor Electrical Engineering, Kettering University

Dr. Mark Thompson, Professor Electrical Engineering, Kettering University

Dr. Kenneth Kaiser, Professor Electrical Engineering, Kettering University

HOW TO USE THIS MANUAL

To locate what you are looking for:

1. The text of this manual is divided into sections.
2. The section titles are listed in the Table of Contents.
3. Holding the manual as shown will allow you to find the first page of the section easily.
4. The contents are listed on the first page of each section to help find the item and page you need.

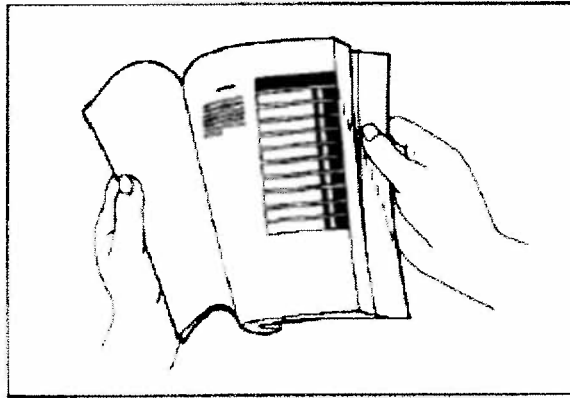
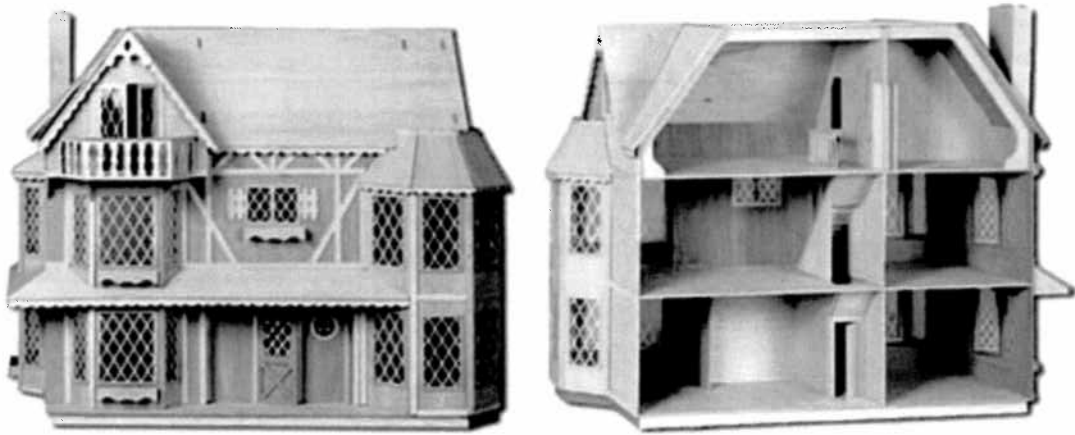


TABLE OF CONTENTS

HOUSE DESCRIPTION	PAGE 1
CIRCUIT DIAGRAM	PAGE 3
PARTS INFORMATION	PAGE 5
INPUT / OUTPUT INFORMATION	PAGE 7
OPERATION SUMMARY	PAGE 8
APPENDICES	
PROGRAM CODING	PAGE 9

HOUSE DESCRIPTION



For this project, we used the Harrison Dollhouse kit manufactured by Greenleaf Dollhouses. The specifications are as follows:

GREENLEAF DOLLHOUSES

436 LAKE ROAD
SCHENEVUS, NY 12155
UNITED STATES OF AMERICA
Phone: (800) 253-7150
Fax: (607) 638-9076

Web Link: <http://www.greenleafdollhouses.com/>

Manufacturer's Part #: 8006

Manufacturer's UPC#: 73605208006

Weight 32 lbs.

Shingles: 5 bags required (rectangular suggested)

Optional Stucco: 5 bottles

Set-up: 37 ¼" W x 21 1/8" D x 31" H

Kit size: 36 ½" W x 24" D x 3" H

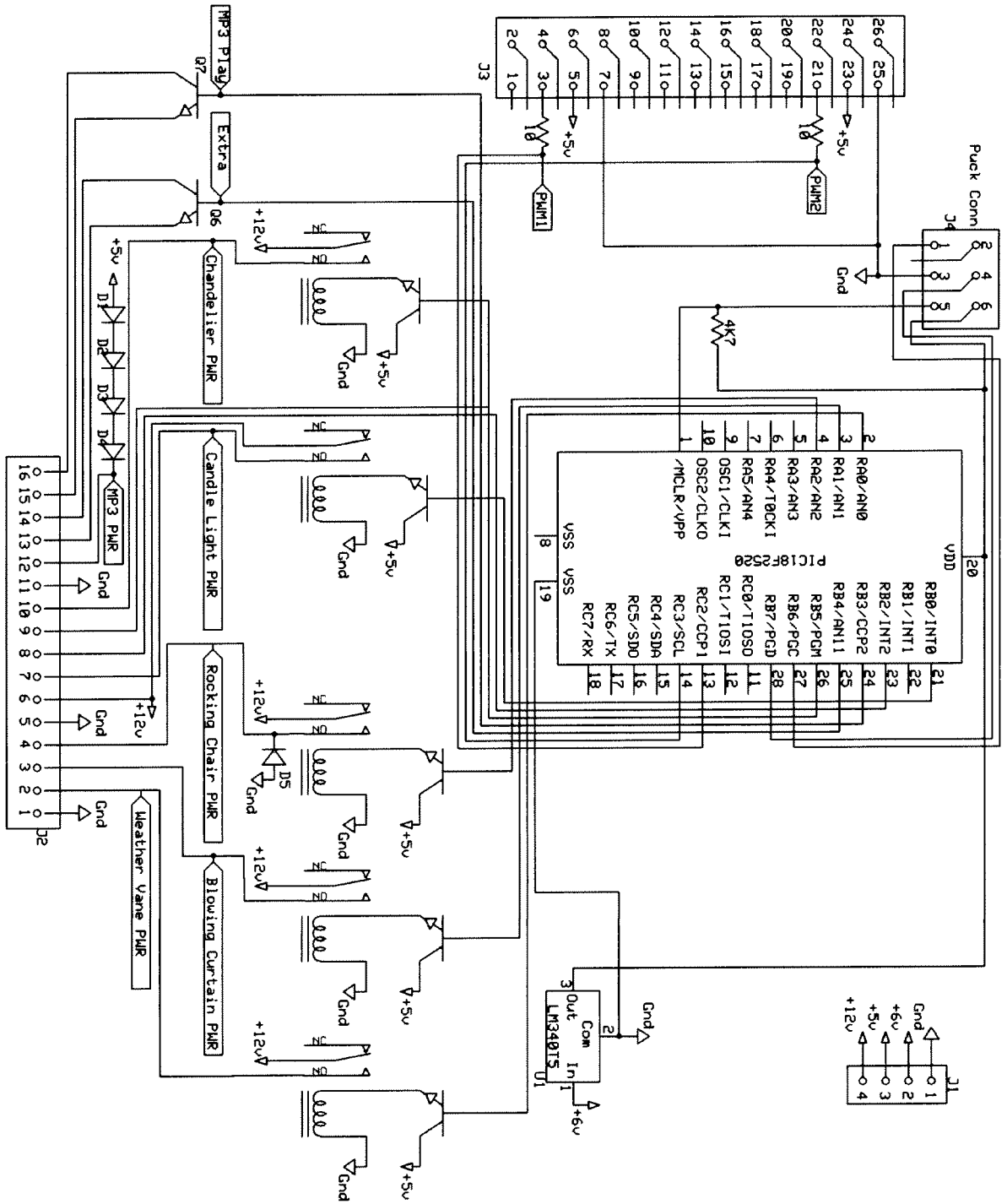
Material: Wood

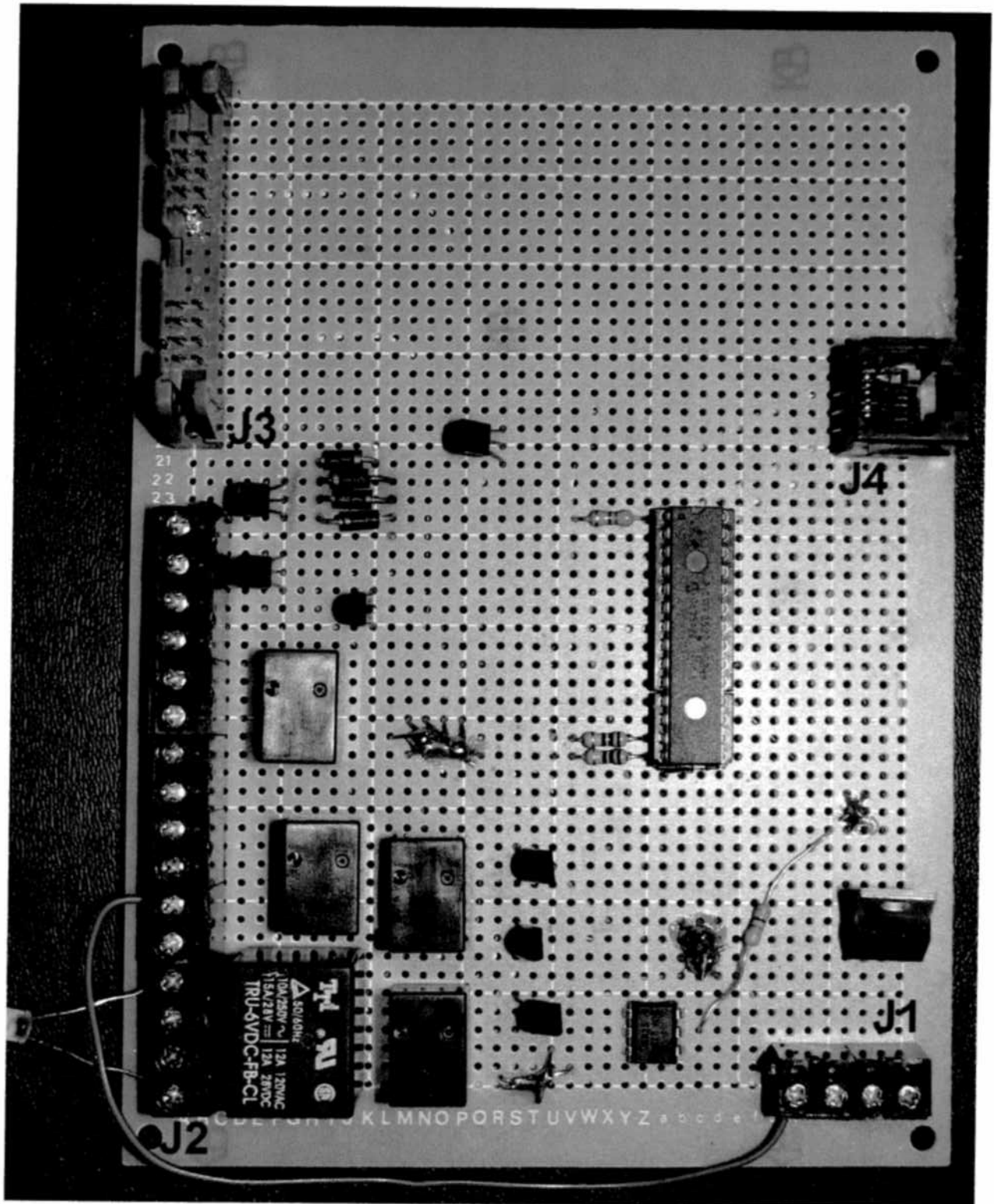
Assembly time is approximately 15 hours

Country of Origin: United States of America

This dollhouse kit does not come with paint, furnishings, or other decorations. Age recommendations: Assembly - 12 and up. Assembly with parent - 7 and up. Finished houses are not recommended for young children.

CIRCUIT DIAGRAM





PARTS INFORMATION

Component #	Component Type	Component Availability
D1	Generic Diode	Kettering Crib
D2	Generic Diode	Kettering Crib
D3	Generic Diode	Kettering Crib
D4	Generic Diode	Kettering Crib
D5	Generic Diode	Kettering Crib
J1	Screw Terminal Connector	Kettering Crib
J2	Screw Terminal Block	RadioShack
J3	Ribbon Cable Connector	Kettering Crib
J4	Puck Connector	Kettering Crib
Q1	2N2222A	Kettering Crib
Q2	2N2222A	Kettering Crib
Q3	2N2222A	Kettering Crib
Q4	2N2222A	Kettering Crib
Q5	2N2222A	Kettering Crib
Q6	2N2222	Kettering Crib
Q7	2N2222	Kettering Crib
R1	10 Ω	Kettering Crib
R2	10 Ω	Kettering Crib
R3	4.7 k Ω	Kettering Crib
RLY1	6 V Relay	TRU-6VDC-FB-CL
RLY2	5 V Relay	OUAZ-SS-105D
RLY3	5 V Relay	OUAZ-SS-105D
RLY4	5 V Relay	OUAZ-SS-105D
RLY5	5 V Relay	OUAZ-SS-105D
U1	LM340T5	Kettering Crib
U2	PIC18F2520	PN @ Microchip.com
	10 conductor ribbon cable	Kettering Crib
	Perforated Circuit board	RadioShack
	Misc. ribbon cable connectors	Kettering Crib
	22awg solid core wire	Kettering Crib
	28 pin IC socket	Kettering Crib
	Futaba S3000 Servo	Rider's Hobby
	Futaba S3000 Servo	Rider's Hobby
	CPU Power Supply	DigiLink
	6 V Generic AC/DC adapter	Kettering Crib
	12 V CPU Cooling Fans	RadioShack
SP-37	12 V micro-mini solenoid	www.electromechanisms.com
	12 V Dollhouse lights	Rau's Country Store, Frankenmuth
	The Harrison Dollhouse	www.greanleafdollhouses.com
	Rectangular cedar shingles	Rau's Country Store, Frankenmuth
	Brass Weathervane	Rau's Country Store, Frankenmuth
	White Rocking Chair	Rau's Country Store, Frankenmuth

Misc. Dollhouse Accessories Rau's Country Store, Frankenmuth
Rustoleum Heirloom White Paint Home Depot
Rustoleum Espresso Brown Paint Home Depot

INPUTS AND OUTPUTS

Inputs

The dollhouse uses no inputs. When power is supplied, the routine is reset and will run until power is no longer supplied.

Outputs

PWM

The Futaba S3000 servo uses a pulse between 20 milliseconds and 30 milliseconds with a duty cycle of one millisecond to two milliseconds. The duty cycle of one millisecond represents a full counterclockwise position, with 1.5 milliseconds represents the center position and two milliseconds represents the full clockwise position. The PIC18F2520 has a maximum of two PWM outputs.

12 V Motors and Lights

Digital outputs on the PIC are used to activate NPN transistors which energize the coils on the five volt DC relays.

12 V Solenoid

The solenoid output is nearly identical to that of the motors with the exception of the diode D5, which is critical to the operation of the PIC. The solenoid was modified by adding a new spring with more stiffness to avoid a binding of the mechanism.

5 V LED

The five volt LED is powered directly from the PIC.

MP3 Player Power

Four diodes are used to reduce the five volt supply to a level that is safe for the MP3 player which was designed to run on a single AAA battery.

MP3 Player Play

Wires are attached to each contact of the play/pause pushbutton switch on the MP3 player. This allows for the play/pause function to be controlled by the PIC.

OPERATION

Startup

- When power is supplied to the circuit, the software positions each servo in its starting position.
- The door will close if it was open, and the man will lie down if he was sitting. This ensures that the routine will function correctly if the power is interrupted at any point other than at the exact end of the routine.
- Then the MP3 player turns on with a prolonged activation of the play/pause button.

Main Routine

- At this point, the software enters a loop that will continue until power is interrupted.
- First, the play function of the MP3 player will be activated to start the sound file.
- The front door will open after an initial delay.
- Next, the weather vane will spin followed by the curtains blowing by the cooling fan.
- Afterwards, the cat will be illuminated by an LED and the solenoid for the rocking chair will be activated at the same time.
- The next action will be the man will sit up and then lay back down after the specified time delay.
- Finally, the candle in the top room will flicker and eventually go out, followed by the front door closing.
- The actions/lights loop is a few seconds longer than the sound file. This ensures that when the loop restarts, the sound file has come to a stop. This allows the sound file to be restarted at the beginning with a single activation of the play/pause button.

APPENDICES

PROGRAM CODE

```

//=====
//
// Filename: dollHouse.c
//=====
//
// Author: Brandon Koenigsknecht
// School: Kettering University
// Revision: 1.00
// Date: 12/10/2006
//=====
//
// Compiled using MPLAB-C18 V1.00.31
// Include Files: P18F2520.H
//      pwm.h
//      p18cxxx.h
//=====
//
// This code is for EE Capstone project "Haunted Doll House" The PICMicrocontroler
// (P18F2520) is used to trigger specific events to occur durning specific timing
// durations.
//
//=====
#include <P18F2520.h>
#include <pwm.h>
#include <p18cxxx.h>
//=====
//=====
//Delay Subroutines, Each routine was originally written for specific delay in time
//which needed to occur in order to synchronize events with the sound track. Some delays
were
//reused in the interest of time and debugging.

void delay (void)
//Originally Standard Delay Length
{
    unsigned int i;
    for (i = 0 ; i < 2500 ; i++)
        ;
}

void delayc (void)

```

```

//Cat Delay
{
    unsigned int c;
    for (c = 0 ; c < 1750 ; c++)
        ;
}

void delaya (void)
//Delay Inbetween Actions
{
    unsigned int a;
    for (a = 0 ; a < 800 ; a++);
}

void delayo (void)
//Delay Actions are on
{
    unsigned int o;
    for (o = 0 ; o < 4000 ; o++);
}

void delays (void)
//The Delay length that a Single is output on the PWM ports
{
    unsigned int s;
    for (s = 0 ; s < 545 ; s++);
}

void delayv (void)
//Weather Vane Delay
{
    unsigned int v;
    for (v = 0 ; v < 3250 ; v++);
}

void delaym (void)
//Music Delay
{
    unsigned int m;
    for (m = 0 ; m < 150 ; m++);
}

void delayd (void)
//dead guy Delay
{
    unsigned int d;

```



```
        for (d = 0 ; d < 1000 ; d++);  
    }
```

```
void delayw (void)  
//wait Delay for music  
{  
    unsigned int w;  
    for (w = 0 ; w < 900 ; w++);  
}
```

```
void delayf (void)  
//flicker Delay  
{  
    unsigned int f;  
    for (f = 0 ; f < 20 ; f++);  
}
```

```
//=====
```

```
=====
```

```
//
```

```
// main routine. The main routine contains all action code as well as "setup code"
```

```
// The setup code changes the PIC clock frequency, clears all digital outputs,
```

```
// puts our PWM outputs in a "start" position, and resets MP3 Player
```

```
// MP3 Player requires a length of connectivity to operate
```

```
//
```

```
//=====
```

```
=====
```

```
void main(void)  
{  
//Variable g is used in flickering actions at end  
    unsigned int g;
```

```
//=====
```

```
=====
```

```
//
```

```
// Setup Code
```

```
//
```

```

//=====
=====

//Sets PIC Oscillator Bits to
//Sets clock speed to 31.7 KHz
    OSCCONbits.IRCF0 = 0;
    OSCCONbits.IRCF1 = 0;
    OSCCONbits.IRCF2 = 0;

//Configures Ports A and B as Digital outputs
    TRISA = 0;
    TRISB = 0;

//Clears Ports A and B
    PORTA = 0;
    PORTB = 0;

//Resets Servo1 (The dead guy that rises) to prone position
    OpenPWM1(500);
    SetDCPWM1(80);
    delays();
    ClosePWM1();
    delays();

//Resets Servo2 (The Front Door) to a Closed Position
    OpenPWM2(500);
    SetDCPWM2(80);
    delays();
    ClosePWM2();
    delays();

//Turns on or resets MP3 Player to beginning of track
//Delays were chosen from those already written in the interest of time

    LATBbits.LATB4 = 1;
    delaya();
    LATBbits.LATB4 = 0;
    delay();

//=====
=====

```

```

//
// Main Loop starts here. This is where the routine will repeat code if left on
// for an extended period of time. Also contains all action and mp3 play play functions
// LED and light outputs on PortB (5), Motor Control on PORTA (5), servo Control on
// PWM Ports (2)
// Order of Actions: (1) Door Opens (PWM2), (2) Weather Vane Spins (PORTA 0), (3)
// Curtain Blowing (PORTA 1),
// (4) Illuminated Cat (PORTB 2), (5) Rocking Chair rocks (PORTA 2), (6) Dead guy rise
// and sit (PWM1),
// (7) Chandlier Flicker (
//
//=====
=====

    while(1){

//MP3 Player Play function
//Delay was specifically written for this function
        LATBbits.LATB4 = 1;
        delaym();
        LATBbits.LATB4 = 0;

//Upstairs Light comes on, Used for "Candle Flicker" at end
        LATBbits.LATB0 = 1;

//delay to synchronize MP3 Player music with the actions in routine
//Delay was specifically written for this function
        delayw();

//1st Action, Door open and close, 1st Mechanical
//Door opens at beginning, and closes near end

//Door Open
        OpenPWM2(418);
        SetDCPWM2(20);
        delays();
        ClosePWM2();
        delays();

//Delay before weather vane spins.
//Delay was specifically written for this action
        delayv();

```

```

//2nd Action, Weather Vane, 2nd Mechanical
//Delay was chosen from those already written in the interest of time
    LATAbits.LATA0 = 1;
    delays();
    LATAbits.LATA0 = 0;

//delay inbetween weather vane spinning and curtain blowing
//Delays were chosen from those already written in the interest of time
    delayv();
    delayo();

//3rd Action, Curtain Fan, 3rd Mechanical
//Delay was chosen from those already written in the interest of time
    LATAbits.LATA1 = 1;
    delayo();
    LATAbits.LATA1 = 0;

//delay inbetween curtian blowing and cat/rocking chair
//Delay was specifically written for this action
    delayc();

//4th Action, Illuminate Cat on roof... LATB2
//5th Action, Rocking Chair, 4th Mechanical...LATA2
//PWM output was included because Servo1 was recieving interference from solenoid
//so PWM was included to "reset" the servo before the servo action happened

//Turn on LED to illuminate Cat on roof
    LATBbits.LATB2 = 1;
//turn on solenoid to cause rocking chair to rock
//Puts solenoid in an extended position
    LATAbits.LATA2 = 1;

//Reset Dead guy
//a longer delay was used for the length that the solenoid was extended
//the is a delay in which the solenoid will react to an output
    OpenPWM1(500);
    SetDCPWM1(80);
    delaym();
    ClosePWM1();

//turn off solenoid, puts solenoid in retracted position to allow rocking
//chair to rock freely
    LATAbits.LATA2 = 0;

//Delays were chosen from those already written in the interest of time
    delayd();

```

```

        delayv();

//Turns cat LED off
        LATBbits.LATB2 = 0;

//delay inbetween Cat/Rocking Chair Actions.
//Delay was chosen from those already written in the interest of time
        delaym();

//6th Action, Dead Rising, 5th Mechanical
//Sits up dead guy
        OpenPWM1(418);
        SetDCPWM1(20);
        delays();
        ClosePWM1();
        delaya();

//Delay between guy sitting and laying
//Delay was chosen from those already written in the interest of time
        delayc();

//Lay down Dead guy
        OpenPWM1(500);
        SetDCPWM1(80);
        delays();
        ClosePWM1();

//Delay inbetween dead guy moving and lightning flicker
//Delays were chosen from those already written in the interest of time
        delaym();
        delays();

//7th Action, lightning flicker
//Delays were written specifically for flickering action
        for (g = 0 ; g < 30 ; g++){
            LATBbits.LATB5 = 1;
            delayf();
            LATBbits.LATB5 = 0;
            delayf();
        };//close for loop

//Delay inbetween lightning and door closing
//Delay was chosen from those already written in the interest of time
        delayo();

```

```
//Door Close  
OpenPWM2(500);  
SetDCPWM2(80);  
delays();  
ClosePWM2();  
  
//Delay inbetween Door Close and Upstairs candle going out  
//Delay was chosen from those already written in the interest of time  
delays();  
  
//8th Action, Candle Flicker  
//Delays were chosen from those already written in the interest of time  
  
    for (g = 0 ; g < 10 ; g++){  
        LATBbits.LATB0 = 1;  
        delaym();  
        LATBbits.LATB0 = 0;  
        delayf();  
    };//close for loop  
  
//Delays at end so that the MP3 will automatically stop at the end of the track  
    delayo();  
    delay();  
  
    //end main while loop  
//end main routine
```