

Gadget Freak Case #161: Golfers Go Green with Envy

Why putt golf balls into a paper cup when you can have your own own variable-terrain putting green? The course designed by Mark Giannasi, Aaron Gaylord, Brody Collins, Thomas Boileau and Maiki Vlahinos offers players three cups and three skill levels. The most advanced level raises two platforms that change the terrain of the course--and the challenge that golfers face. But even with only three "cups," you can still shout "Fore!"

This Gadget Freak project serves as a model you can use to create your own course that could include different terrain features, cups, and controls. The builders, all students in the Department of Mechanical Engineering at Colorado State University, took advantage of a variety of microcontrollers, electronic components, motors, and batteries to create this project. Watch their video that illustrates course operations at: http://www.designnews.com/video/Gadget_Freak/4379-Golfers_Go_Green_with_Envy.php.

A control panel mounted on a post lets players turn on the system and choose the course difficulty--Easy, Medium, or Hard. The control panel includes a 2-line LCD alphanumeric display, a 12-button keypad, and indicators and switches. A speaker creates various sound effects.

The Easy course comprises a flat terrain and the right-most cup. But when a player chooses the Medium-difficulty course, a motor under the course turns a threaded rod that moves a wedge to lift a small round platform. As a result, a small mound appears in the midst of the course. Medium-level players aim for the cup on the left side of the course. If a player chooses the Hard course, two motor-driven platforms raise the round mound and a plateau on the course, and the player must aim for the middle cup underneath the windmill hazard. Players can choose to put a ball over the bridge, instead.

LEDs illuminate the target cup for each difficulty level. Putting a ball into the proper target cup causes the console to play musical notes and display a congratulatory message. If your ball goes in the wrong cup, the console's display notes, "Wrong Hole Try Again," followed by a second message to aim for the illuminated cup. In all cases, balls that enter any cup return to the front of the course through a series of pipes beneath the green.

You can play this course at "night," too. Turn off the room lights and a photosensor hidden in the mountains detects darkness and causes an operational amplifier to turn on LEDs that light the course. This lighting feature operates independent of the system's microcontrollers.

The following sections explain the operation of each subsystem. This document also includes photographs, schematic diagrams, a complete bill of materials, and code listings for the microcontrollers. This Gadget Freak Case does not include step-by-step construction instructions because most builders would likely adapt the design to available space and the terrain and "target" features they want.

LEDs and Liquid-Crystal Display

This project used many LEDs that serve as indicators, create accent lighting, and illuminate the course. A 2-line liquid-crystal display (LCD) displays messages and instructions for players who can choose the level of difficulty for the course and receive visual feedback of their performance.

Audio Output

The golf-course console included a speaker that produces a software-generated melody triggered by a switch in each cup. The tune signifies a player has successfully completed the chosen course.

Manual Data Input

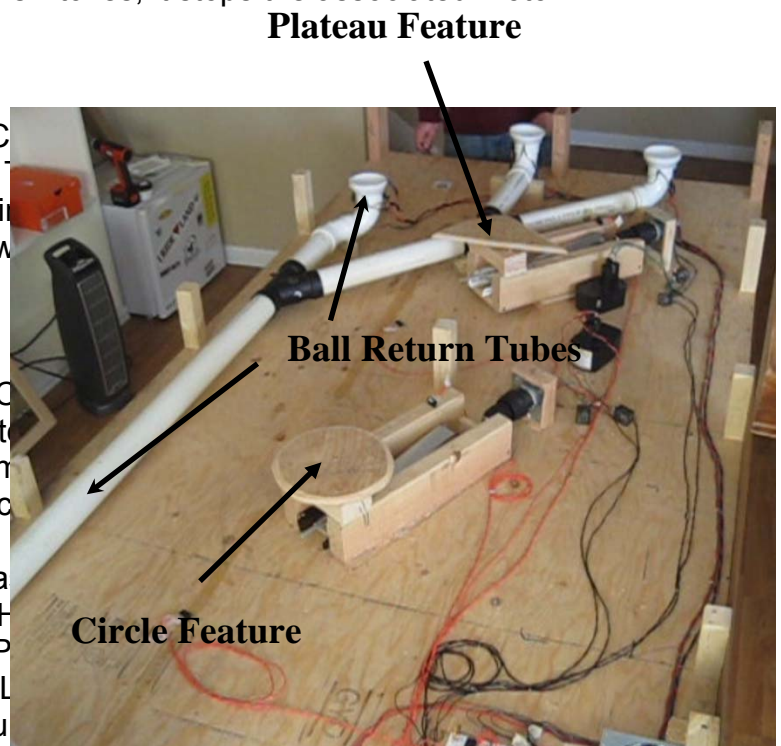
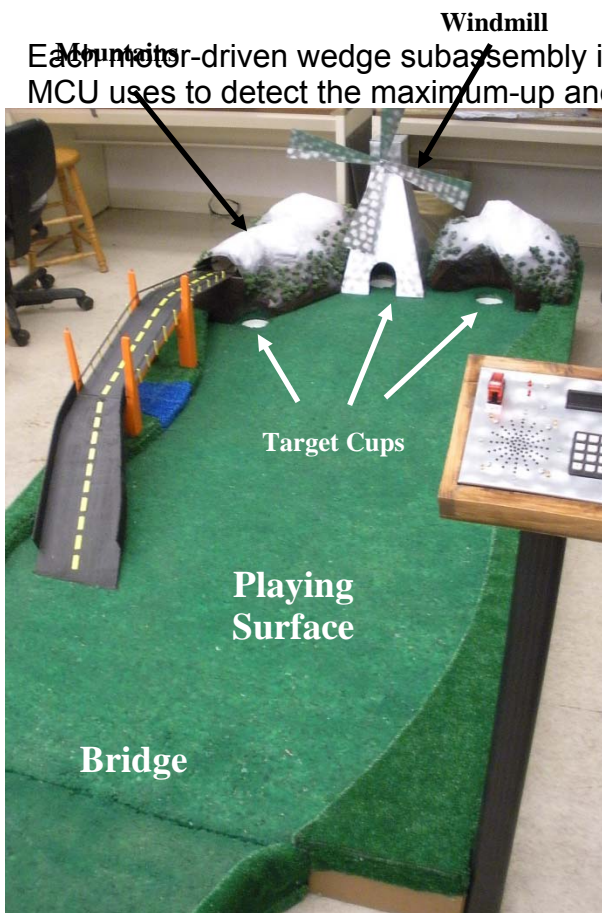
A 12-key numeric keypad with normally-open key-switch contacts lets a player choose a level of difficulty. In this project, the designers included three levels of difficulty, but you can implement more keyboard-selected options such as additional moveable terrain features, time limits, and so on.

Sensor Inputs

Each of the three cups includes a switch that a golf ball can actuate. When a player sinks a ball into the proper target cup, the switch alerts a master microcontroller that sends data to the LCD and alerts other MCUs to transmit sounds to the speaker and control LEDs. The master MCU also resets the course features for the next player. When a ball enters a cup other than the illuminated target cup, its switch alerts the master MCU to send messages to the LCD to remind the player to aim for the correct hole.

A photocell built into the "mountains" detects ambient light and causes a circuit to turn on high-intensity LEDs around the playing field for "night golf."

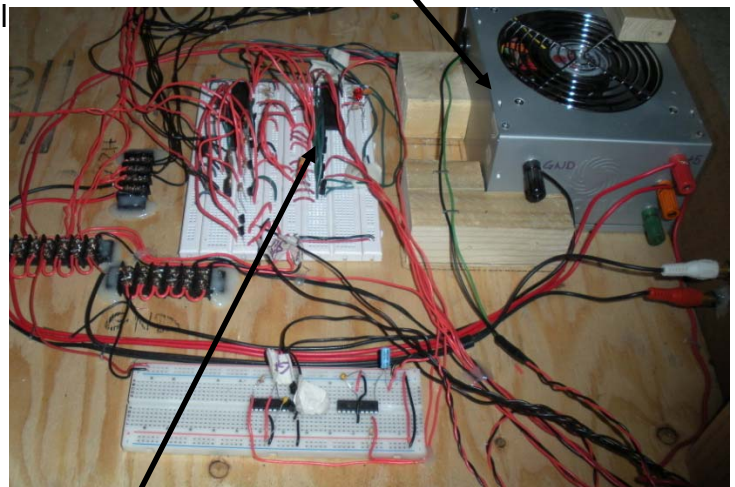
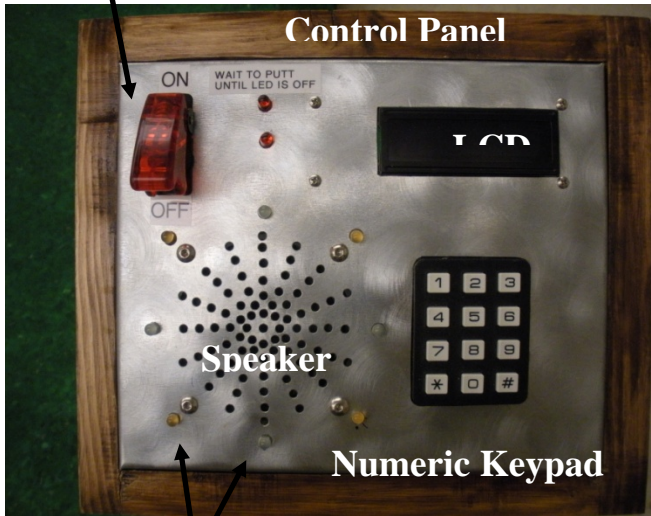
Each motor-driven wedge subassembly includes limit switches that the motor-control MCU uses to detect the maximum-up and home (down) position for each wedge. When the switches, it stops the associated motor.



2. The second PIC16F884 MCU controlled the motors and monitors the limit switches on the screw/wedge mechanisms.

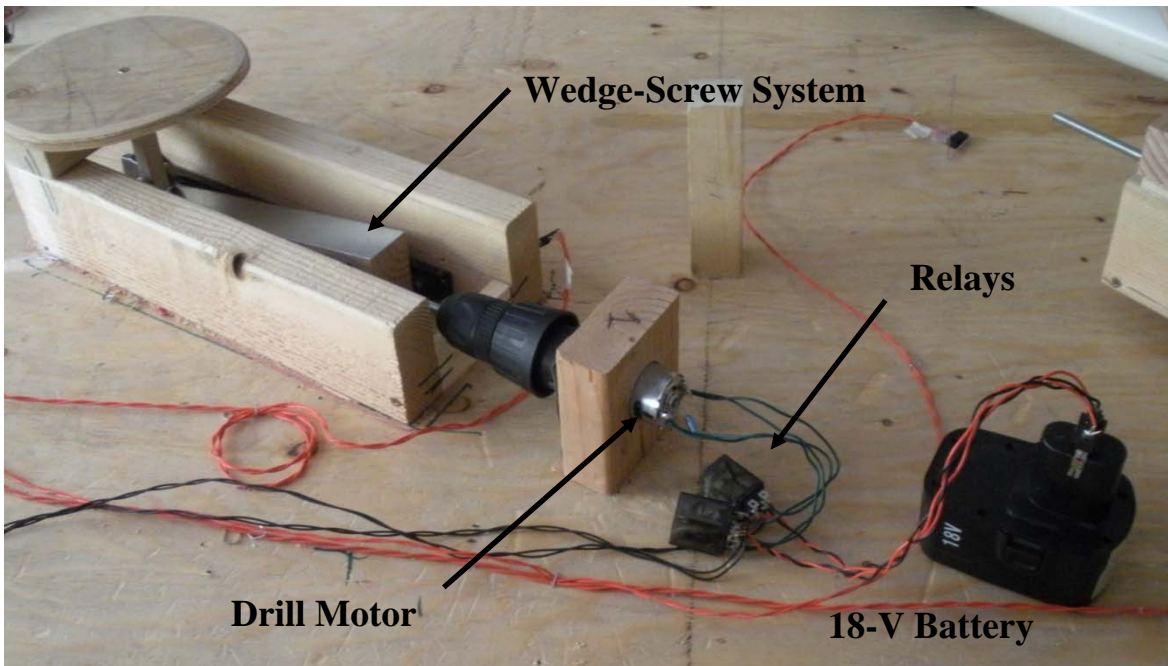
Power Switch The PIC16F88 MCUs handle the bridge-lighting, score LEDs, and generate the sounds. The use of individual MCUs programmed for specific tasks simplified the

Power Supply



Flashing LED (8 places)

Master PIC MCU



Operational Amplifier

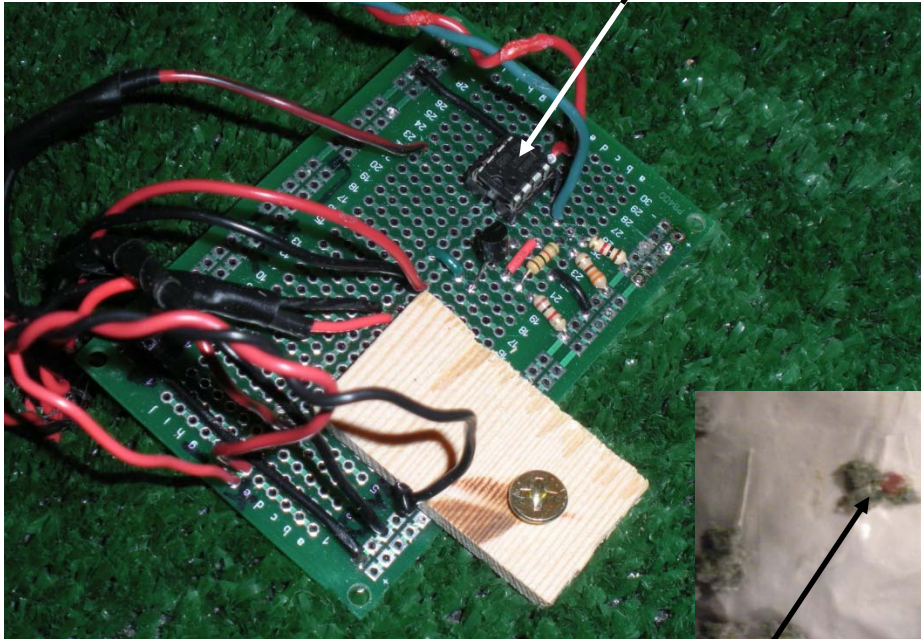


Photo Cell hidden in Mountain

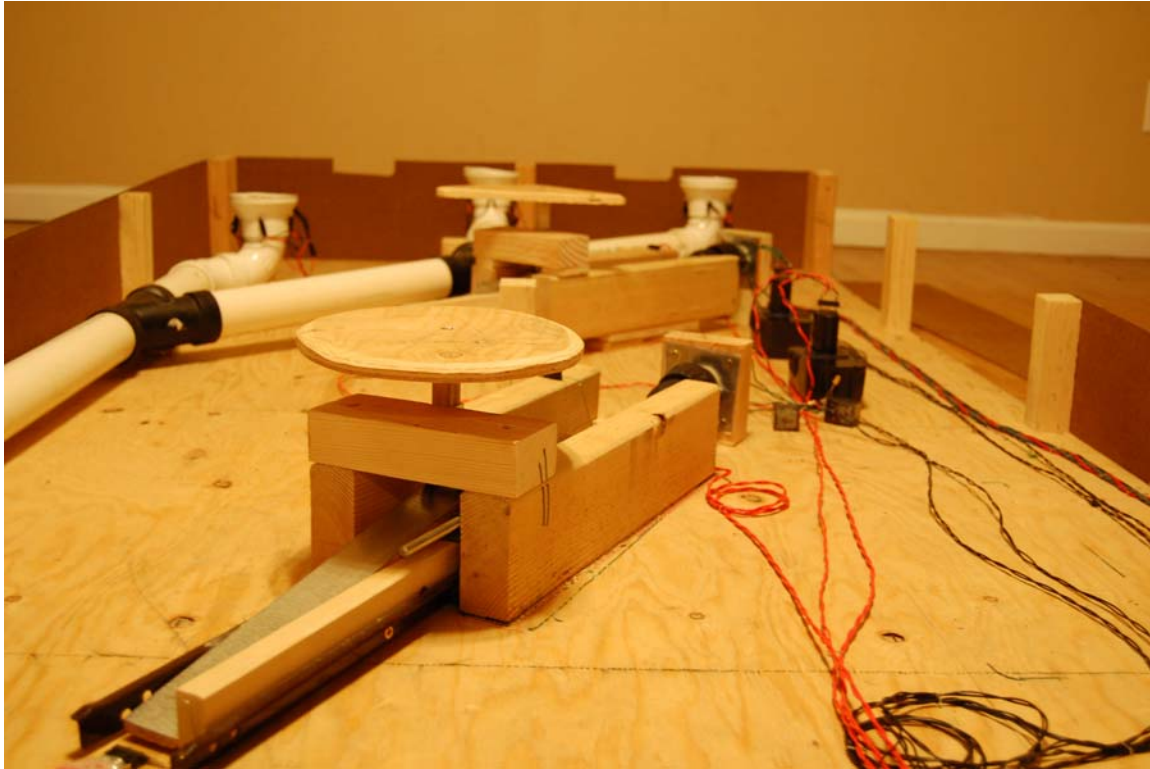
More Golf Course Photos



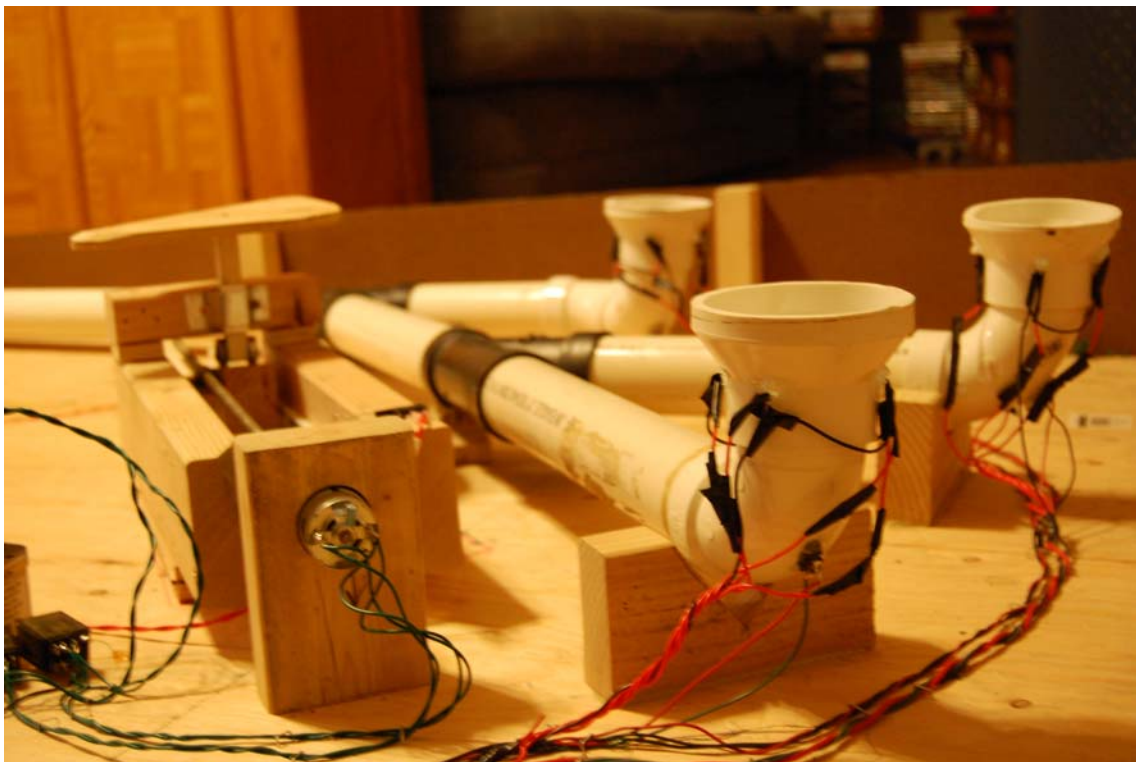
The three illuminated "target" cups.



Course from putter's perspective.



The underlying terrain-lifting mechanisms and ball-return tubes.



Another view of the mechanisms and cups. Note the illumination LEDs around the funnel for each cup and the small switch placed in the elbow.



Power supply and breadboards used to create the control circuits.



Overall view of the golf course and control panel. The "assembly" below the control panel



Golf course with two terrain features in raised positions.

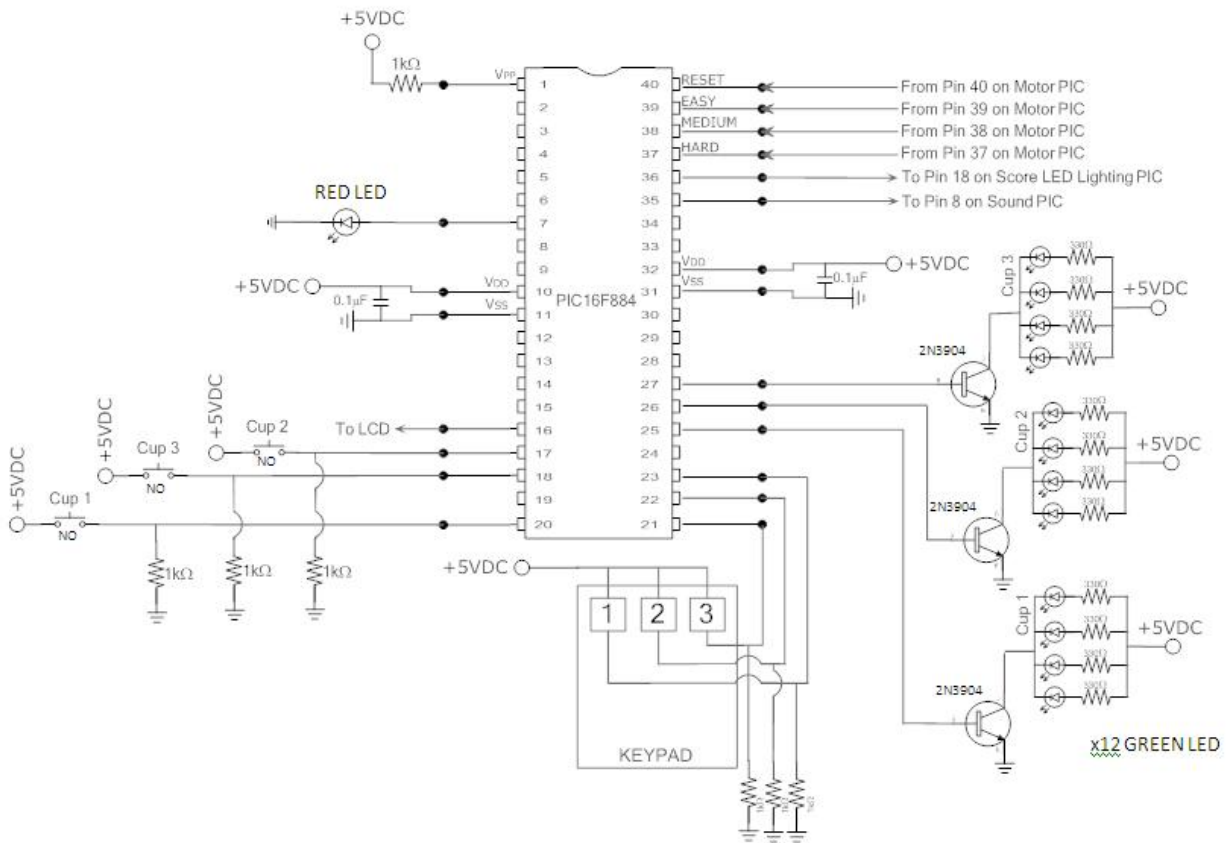
Gadget Freak Case #161 Bill of Materials

Amt	Part Description	Allied Part #
3	PIC16F88 Microcontroller	383-0496
2	PIC16F884 Microcontroller	383-1232
7	Switch, SPDT Leaf Lever (Used for wedge position sensors & cup trigger sensors)	821-0482
1	Switch, SPDT (Power on)	821-0563
1	Keypad, 12 digit	948-0014
4	Relay, DPDT, 3A, 12V	886-2250
4	TIP3055 Power Transistor	568-1031
13	2N3904 NPN Transistor	411-0029
1	2N3906 PNP Transistor	568-0293
7	0.1 μ F Capacitor, 50V	852-1170
17	1K Ω Resistor 1/4W	296-4741
37	330 Ω Resistor 1/4W	296-4764
10	LED, High Intensity, White	749-3330
7	LED, Red	405-0011
4	LED, Yellow	405-0137
16	LED, Green	405-0017
1	LF356N Op-Amp, JFET Input	288-0356
1	Pinned Photo Cell	980-0010
3	Solderless Breadboard	237-0015
1	16 x 2 Serial LCD (Sparkfun Electronics)	N/A
2	18V Drill (Harbor Freight)	N/A

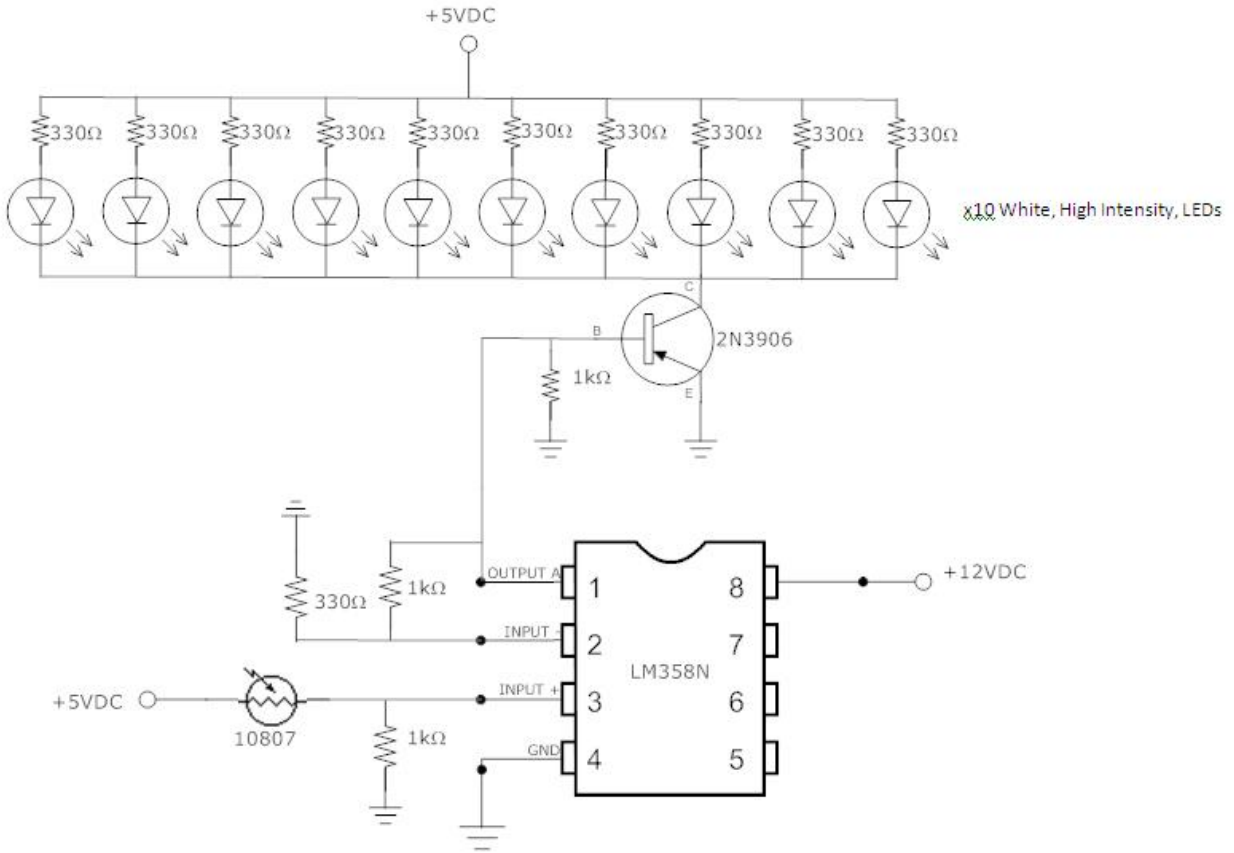
1	Computer Power Supply (±12V, 5V, 3.3V)	N/A
3	4' x 8' 3/4" Plywood	N/A
5	2-inch x 4-inch x 8-foot dimensional lumber	N/A
1	1/4-inch Particle Board (Trim)	N/A
2	Carpet Padding (8-feet x 4-feet)	N/A
1	3-inch-diameter PVC Tubing (16-feet plus fittings and glue)	N/A

Gadget Freak Case #161 Schematic Diagrams

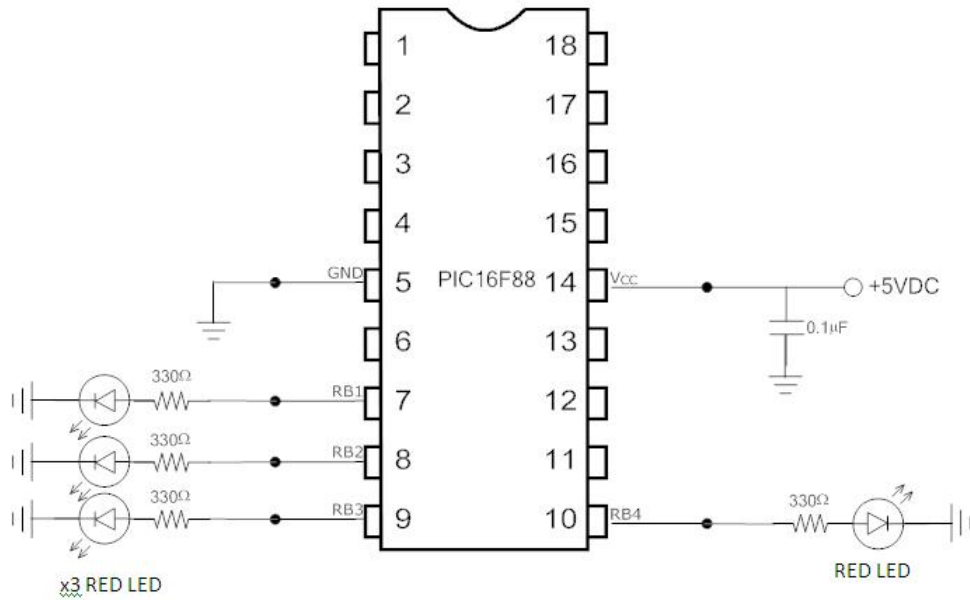
Master Controller



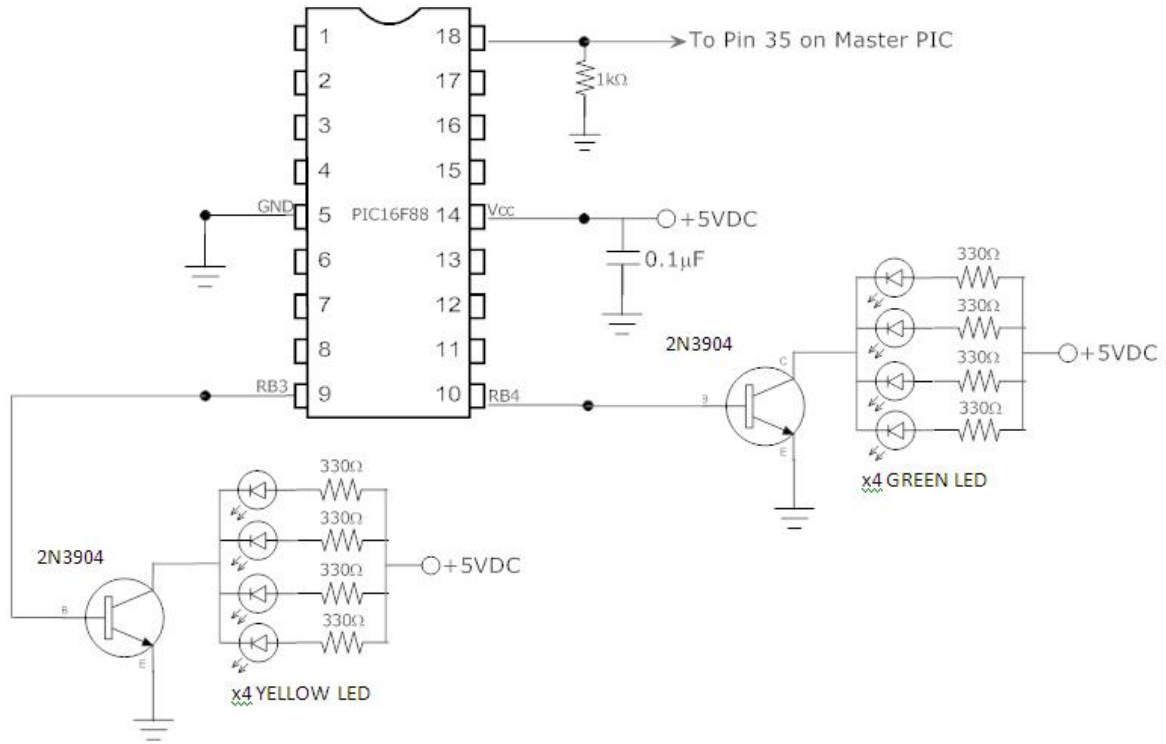
Motor Controller



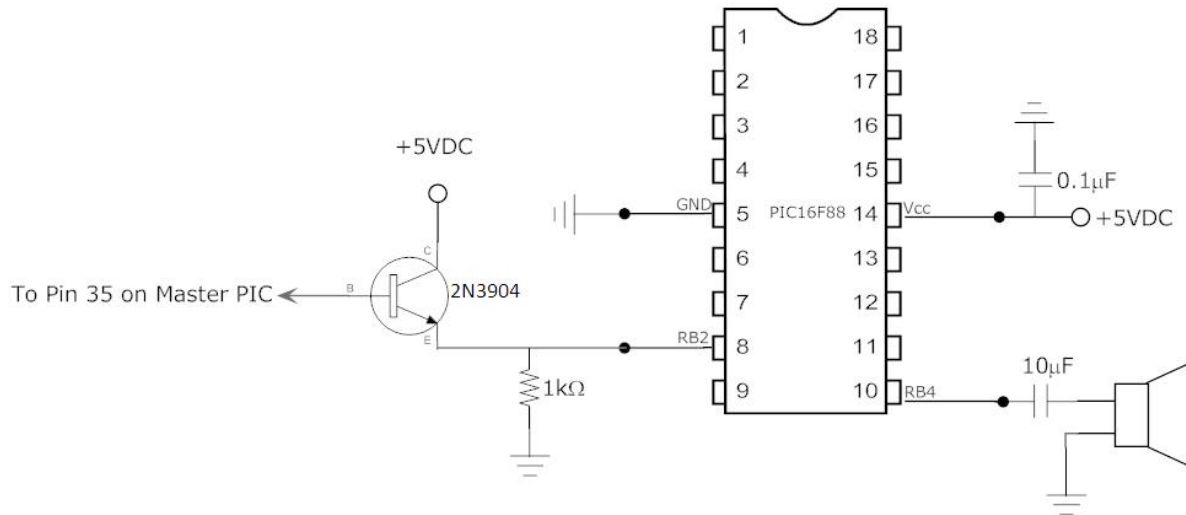
Bridge Lighting



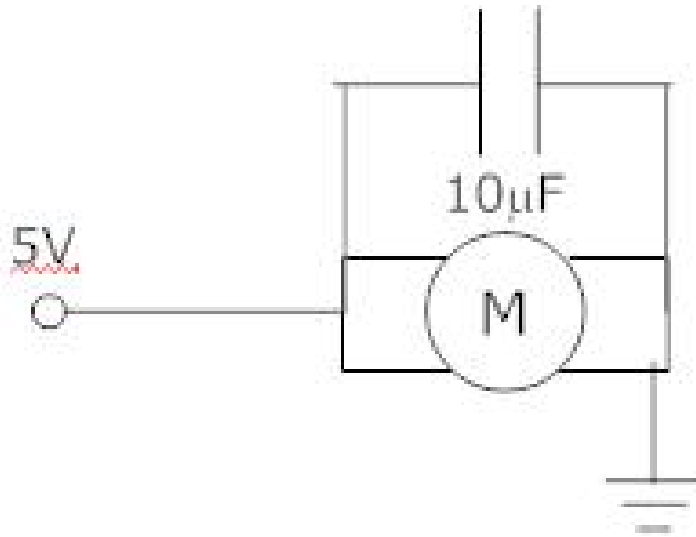
Score LEDs



Sound Generator



Windmill Motor (always on)



Gadget Freak Case #161 Code Listings (for PICBasic PRO compiler).

a. Master PIC Code

```

DEFINE OSC 8
OSCCON.4 = 1    'Sets the internal oscillator frequency to 8 MHz
OSCCON.5 = 1
OSCCON.6 = 1
ansel = 0
anselH = 0000000    'Turns off analog to digital conversion.

'DEFINE VARIABLES
RESETOUT VAR PORTB.7
EASY VAR PORTB.6
MEDIUM VAR PORTB.5
HARD VAR PORTB.4
SLCD VAR PORTC.1
ILED VAR PORTA.5
BUT3 VAR PORTD.2
BUT2 VAR PORTD.3
BUT1 VAR PORTC.4
CUP1S VAR PORTD.1
CUP2S VAR PORTC.2
CUP3S VAR PORTC.3
LIGHTS VAR PORTB.2
SOUND1 VAR PORTB.3
CUP1LIGHT VAR PORTC.6
CUP2LIGHT VAR PORTC.7
CUP3LIGHT VAR PORTD.4

'DEFINE CONSTANTS
INCLUDE "modedefs.bas"
LOW RESETOUT
LOW EASY
LOW MEDIUM
LOW HARD

```

```
LOW SLCD
LOW BUT1
LOW BUT2
LOW BUT3
LOW CUP1S
LOW CUP2S
LOW CUP3S
LOW CUP1LIGHT
LOW CUP2LIGHT
LOW CUP3LIGHT
LOW SOUND1
LOW LIGHTS
```

```
'START OF CODE
  LOW ILED
  PAUSE 300
  HIGH ILED
  PAUSE 300
  LOW ILED
  PAUSE 2000 'TO WAIT FOR SLAVE PICS TO START
  SEROUT SLCD,T9600,[$FE,1] 'CLEAR LCD DISPLAY OF ANYTHING
  SEROUT SLCD,T9600,[$FE, 1,"Welcome To..."] 'clear lcd and display
welcome to
  PAUSE 1500 'For 1.5 seconds
  SEROUT SLCD,T9600,[$FE, 1, "   How the", $FE, $C0, "   Green Lies!!"]
'clear display and show how the on first line followed by green lies on
second line
  PAUSE 2000 'for 3 seconds

loop1:
  Serout SLCD,T9600,[$FE,1,"Select Course..."]
  HIGH RESETOUT
  PAUSE 300
  LOW RESETOUT
  PAUSE 1500 'DISPLAY FOR 2 SECONDS
  SEROUT SLCD,T9600,[$FE,1, "ESY   MED   HRD"]
  SEROUT SLCD,T9600,[$FE, $C0, " 1     2     3"]

loop:
  IF (BUT1==1) THEN
    PAUSE 250 ' SWITCH BOUNCE
    HIGH EASY
    PAUSE 300
    LOW EASY
    HIGH ILED 'INDICATOR LED
    PAUSE 300
    LOW ILED
    SEROUT SLCD,T9600,[$FE,1,"   EASY", $FE, $C0, "AIM FOR LIT CUP!"]
    GOTO C1
  ENDIF

  IF (BUT2==1) THEN
    PAUSE 250 ' SWITCH BOUNCE
    HIGH MEDIUM
    PAUSE 300
    LOW MEDIUM
    HIGH ILED
```

```
        PAUSE 300
        LOW ILED
        PAUSE 1000
        SEROUT SLCD,T9600,[$FE,1,"        MEDIUM", $FE,$C0,"AIM FOR LIT
CUP!"]
        GOTO C2
    ENDIF

    IF (BUT3==1) THEN
        PAUSE 250 ' SWITCH BOUNCE
        HIGH HARD
        PAUSE 300
        LOW HARD
        HIGH ILED
        PAUSE 300
        LOW ILED
        PAUSE 1000
        SEROUT SLCD,T9600,[$FE,1,"        HARD", $FE,$C0,"AIM FOR LIT CUP!"]
        GOTO C3
    ENDIF

GOTO loop

C1:
    HIGH CUP1LIGHT
    LOW CUP1S
    LOW CUP2S
    LOW CUP3S
    IF (CUP1S==1)THEN
        HIGH LIGHTS
        HIGH SOUND1
        PAUSE 300
        LOW LIGHTS
        LOW SOUND1
        SEROUT SLCD,T9600,[$FE,1,"NICE PUTT!!!"]
        PAUSE 2000
        LOW CUP1LIGHT
        GOTO LOOP1
    ENDIF

    IF (CUP2S==1)THEN
        GOSUB WRONGHOLE1
    ENDIF

    IF (CUP3S==1)THEN
        GOSUB WRONGHOLE1
    ENDIF

GOTO C1

C2:
    HIGH CUP2LIGHT
    LOW CUP1S
    LOW CUP2S
    LOW CUP3S
    IF (CUP2S==1)THEN
        HIGH LIGHTS
```

```
        HIGH SOUND1
        PAUSE 300
        LOW LIGHTS
        LOW SOUND1
        SEROUT SLCD,T9600,[$FE,1,"NICE PUTT!!!"]
        PAUSE 2000
        LOW CUP2LIGHT
        GOTO LOOP1
ENDIF

IF (CUP1S==1)THEN
    GOSUB WRONGHOLE2
ENDIF

IF (CUP3S==1)THEN
    GOSUB WRONGHOLE2
ENDIF

GOTO C2

C3:
    HIGH CUP3LIGHT
    LOW CUP1S
    LOW CUP2S
    LOW CUP3S
    IF (CUP3S==1)THEN
        HIGH LIGHTS
        HIGH SOUND1
        PAUSE 300
        LOW LIGHTS
        LOW SOUND1
        SEROUT SLCD,T9600,[$FE,1,"NICE PUTT!!!"]
        PAUSE 2000
        LOW CUP3LIGHT
        GOTO LOOP1
    ENDIF

    IF (CUP1S==1)THEN
        GOSUB WRONGHOLE3
    ENDIF

    IF (CUP2S==1)THEN
        GOSUB WRONGHOLE3
    ENDIF

GOTO C3

WRONGHOLE1:
    SEROUT SLCD,T9600,[$FE,1," WRONG HOLE", $FE,$C0, " TRY AGAIN!"]
    PAUSE 2000
    SEROUT SLCD,T9600,[$FE,1," EASY", $FE,$C0,"AIM FOR LIT CUP!"]
    PAUSE 2000
    RETURN

WRONGHOLE2:
    SEROUT SLCD,T9600,[$FE,1," WRONG HOLE", $FE,$C0, " TRY AGAIN!"]
    PAUSE 2000
```

```
SEROUT SLCD,T9600,[$FE,1," MEDIUM", $FE,$C0,"AIM FOR LIT CUP!"]
PAUSE 2000
RETURN
```

WRONGHOLE3:

```
SEROUT SLCD,T9600,[$FE,1," WRONG HOLE", $FE,$C0, " TRY AGAIN!"]
PAUSE 2000
SEROUT SLCD,T9600,[$FE,1," HARD", $FE,$C0,"AIM FOR LIT CUP!"]
PAUSE 2000
RETURN
```

End

b. Motor PIC Code

```
DEFINE OSC 8
OSCCON.4 = 1 'Sets the internal oscillator frequency to 8 MHz
OSCCON.5 = 1
OSCCON.6 = 1
ansel = 0
anselH = 0000000 'Turns off analog to digital conversion.
```

'DEFINE VARIABLES

```
RESETIN VAR PORTB.7
EASYIN VAR PORTB.6
MEDIUMIN VAR PORTB.5
HARDIN VAR PORTB.4
```

```
BS1 VAR PORTC.3
BS0 VAR PORTC.4
BE1 VAR PORTD.1
BE2 VAR PORTD.2
BI1 VAR PORTD.0
BI2 VAR PORTD.3
```

```
PE1 VAR PORTC.2
PE2 VAR PORTC.5
PI1 VAR PORTC.1
PI2 VAR PORTC.6
PS1 VAR PORTC.0
PS0 VAR PORTC.7
```

```
ILED VAR PORTA.0
RLED VAR PORTA.1
```

'DEFINE CONSTANTS
LOW ILED

'START OF PROGRAM
LOW RESETIN
LOW EASYIN
LOW MEDIUMIN
LOW HARDIN

LOW BE1
LOW BE2

```
LOW BI1
LOW BI2
```

```
LOW PE1
LOW PE2
LOW PI1
LOW PI2
```

```
HIGH ILED
PAUSE 500
LOW ILED
```

```
ALWAYS:
```

```
    IF (RESETIN==1) THEN
        GOTO RESET
    ENDIF
```

```
    IF (EASYIN==1) THEN
        HIGH ILED
        PAUSE 1000
        LOW ILED
        LOW EASYIN
    ENDIF
```

```
    IF (MEDIUMIN==1) THEN
        GOTO EASY
    ENDIF
```

```
    IF (HARDIN==1) THEN
        GOTO MEDIUM
    ENDIF
```

```
GOTO ALWAYS
```

```
EASY:
```

```
    LOW MEDIUMIN
    HIGH ILED
    HIGH BE2
    HIGH BI2
    GOTO EASY1
```

```
EASY1:
```

```
    IF (BS1==1) THEN
        LOW ILED
        LOW BE2
        LOW BI2
        GOTO ALWAYS
    ENDIF
    GOTO EASY1
```

```
MEDIUM:
```

```
    LOW HARDIN
    HIGH ILED
    HIGH BE2
    HIGH BI2
    GOTO MEDIUM1
```

MEDIUM1:

```
    IF (BS1==1) THEN
        LOW ILED
        LOW BE2
        LOW BI2
        GOTO MEDIUM2
    ENDIF
    GOTO MEDIUM1
```

MEDIUM2:

```
    HIGH ILED
    HIGH PE2
    HIGH PI2
    GOTO MEDIUM3
```

MEDIUM3:

```
    IF (PS1==1) THEN
        LOW ILED
        LOW PE2
        LOW PI2
        GOTO ALWAYS
    ENDIF
    GOTO MEDIUM3
```

RESET:

```
    LOW RESETIN
    HIGH RLED
    HIGH BE1
    HIGH BI1
    GOTO RESET1
```

RESET1:

```
    IF (BS0==1) THEN
        LOW RLED
        LOW BE1
        LOW BI1
        GOTO RESET2
    ENDIF
    GOTO RESET1
```

RESET2:

```
    HIGH RLED
    HIGH PE1
    HIGH PI1
    GOTO RESET3
```

RESET3:

```
    IF (PS0==1) THEN
        LOW RLED
        LOW PE1
        LOW PI1
        GOTO ALWAYS
    ENDIF
    GOTO RESET3
```

END

c. Score Lighting PIC Code

```
DEFINE OSC 8
OSCCON.4 = 1           'Sets the internal oscillator frequency to 8 MHz
OSCCON.5 = 1
OSCCON.6 = 1

ansel = 0             'Turns off analog to digital conversion. Refer to
                    'Threaded Design Example A.4 p.296-299 of the textbook
                    'for an example of how to configure and use A/D conversion

TRISA = %00000010
TRISB = %00000000

LOW PORTB.2
LOW PORTB.5

'Score Lighting LEDs; waiting for cup switch
loop:
  If (PORTA.2 == 0) Then loop
  If (PORTA.2 == 1) Then
    Pause 250           'switch bounce
    Goto FLASH
  Endif

FLASH:
  HIGH PORTB.2
  PAUSE 250
  LOW PORTB.2
  PAUSE 250

  HIGH PORTB.5
  PAUSE 250
  LOW PORTB.5
  PAUSE 250

  HIGH PORTB.2
  PAUSE 250
  LOW PORTB.2
  PAUSE 250

  HIGH PORTB.5
  PAUSE 250
  LOW PORTB.5
  PAUSE 250

  HIGH PORTB.2
  PAUSE 250
  LOW PORTB.2
  PAUSE 250

  HIGH PORTB.5
  PAUSE 250
  LOW PORTB.5
  PAUSE 250
```

```
        HIGH PORTB.2
        PAUSE 250
        LOW PORTB.2
        PAUSE 250

        HIGH PORTB.5
        PAUSE 250
        LOW PORTB.5
        PAUSE 250

        HIGH PORTB.2
        PAUSE 250
        LOW PORTB.2
        PAUSE 250

        HIGH PORTB.5
        PAUSE 250
        LOW PORTB.5
        PAUSE 250
    Goto loop
End

d. Sound PIC Code

'SOUND
DEFINE OSC 8
OSCCON.4 = 1           'Sets the internal oscillator frequency to 8 MHz
OSCCON.5 = 1
OSCCON.6 = 1

ansel = 0             'Turns off analog to digital conversion.
loop:
    If (PORTB.2 == 0) Then loop
    If (PORTB.2 == 1) Then
        SOUND
        PORTB.4, [100,25,80,21,88,19,95,26,75,50,100,25,80,21,88,19,95,26,75,50,85,20,
        95,25,100,50,105,20,110,20]
        Endif
        Goto loop
End
```

-----End of Gadget Freak Case #161-----