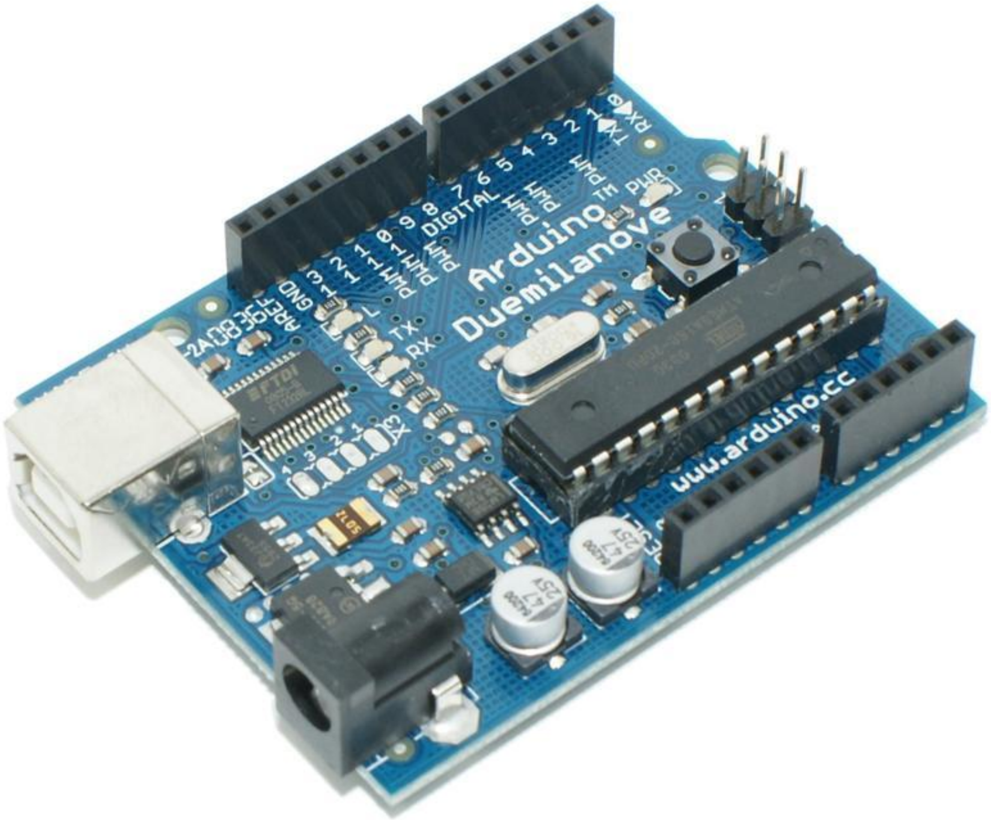
2010

Vishal Kotcherlakota
Ansum Dholakia
ECE 118
6/3/2010



[HOME AUTOMATION USING AN ARDUINO]

Ansum Dholakia, Vishal Kotcherlakota

CONTENTS

Ansum Dholakia, Vishal Kotcherlakota

INTRODUCTION

Welcome to Arduino Home Automation! The system you own is a flexible and powerful tool for automating various actions within your home. Read on, and we'll explain how the system works, and how you can use it.

AN OVERVIEW

In short, our system is a web-based home automation system. We have integrated multiple systems of communication and control together, and have presented them in a single unified interface. Any device on the home's wired or wireless network can access this interface and issue commands—without a need for prior installed software. At last, you can kiss all your remotes good-bye!

SYSTEMS INTEGRATED

There are several systems that we've integrated into this product. We'll be going into detail about them later on, but for now, we'll briefly enumerate and explain them.

X10. The X10 home automation system has been around for 35 years. It is an open industry standard for home automation control using your house's power wiring. Remotes communicate wirelessly with a transciever module plugged into a power outlet. The transciever then relays the codes out over the power wiring using the 120V AC signal as a carrier wave.

Infrared Blasters. The first Infrared Remote was developed in 1977, and since then, has proven to be the control method of choice for many homeowners. As such, every piece of home theater equipment today has a remote eye—a chip that can receive IR signals modulated with a specific frequency.

Apache 2. Apache 2 is an open standards web server, a popular and easy way to host websites on a local network. When a user enters the IP address of a machine running Apache into a web browser, the browser will display a website hosted on the server.

Telnet. Telnet is an open standard for remote communication, typically over IP. It was once used to handle remote logins and to provide command-line services to an end user.

PHP. “PHP: Hypertext Preprocessor”, as it is more formally known, is a server-side scripting language. It is used to build pages that are dynamic, and provide for interaction between a client and a web server. PHP can be used to remotely run tasks and events on a web server from a client.

OUR APPROACH

Our apporoach to this project was to build a 3-layer system of control, beginning with device interfacing, and ending with the end-user.

DEVICE INTERFACING: INFRARED

At the lowest level, we handle device interfacing. With the home theater equipment, this is handled by the placement of IR blasters on the remote eye. IR blasters are nothing more that infrared LEDs connected to a circuit that drives the LED with a control signal.

IR signals are a curious breed. Figure 1 serves as a good starting point for our discussion—the modulating frequency.

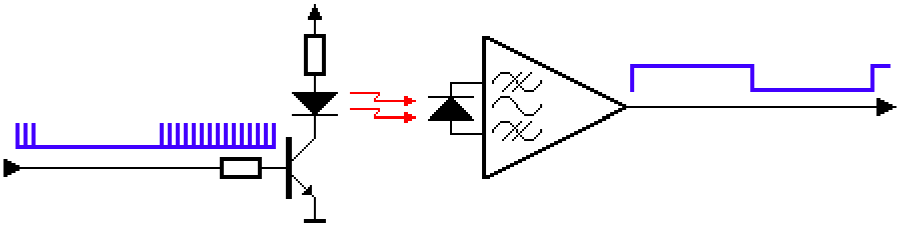


FIG. 1: MODULATING AN IR SIGNAL (HTTP://WWW.SBPROJECTS.COM/KNOWLEDGE/IR/IR.HTML)

It is not enough to turn on an infrared LED to indicate a mark. Every light source emits some infrared radiation, making that portion of the spectrum very crowded. Therefore, we must use modulation to ensure that the device responds only to the IR signal from our controller, and not from our lights. We do this by encoding a mark as a “strobe” of the LED, where the LED blinks rapidly at a specific modulating frequency.

The reciever has a photodiode and a bandpass filter, centered at the modulating frequency. In this way, the chip is able to extract the marks and spaces, which are then fed into logic onboard the device, which will convert the signal into an action (e.g. turning on a device, raising/lowering the volume, etc).

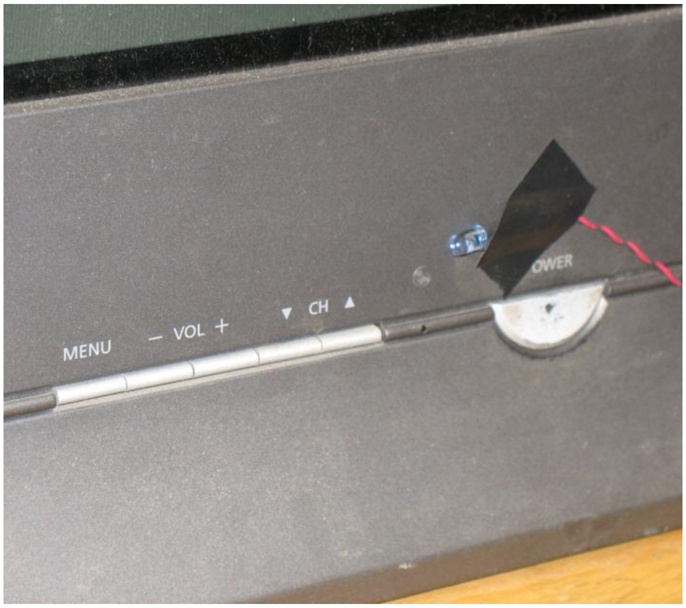


FIG. 2: AN IR BLASTER, WIRED TO OUR TV. THE REMOTE EYE CAN BE SEEN, NEAR THE TIP OF THE LED.

The key, then, is to replicate the IR signal using an LED of our own—enter the IR blaster, shown in Figure 3. The IR LED is set to strobe for intervals as defined in our Arduino code.

It's important to note that while the Arduino timings are imprecise (with a resolution of approximately 100 microseconds), IR receivers and control logic circuits are designed with significant tolerances in mind, and are able to accept our mimicry of the original remote's signals.

DEVICE INTERFACING: X10

X10 is a much more difficult system to interface with, so the discussion of this topic will be kept brief and high-level.

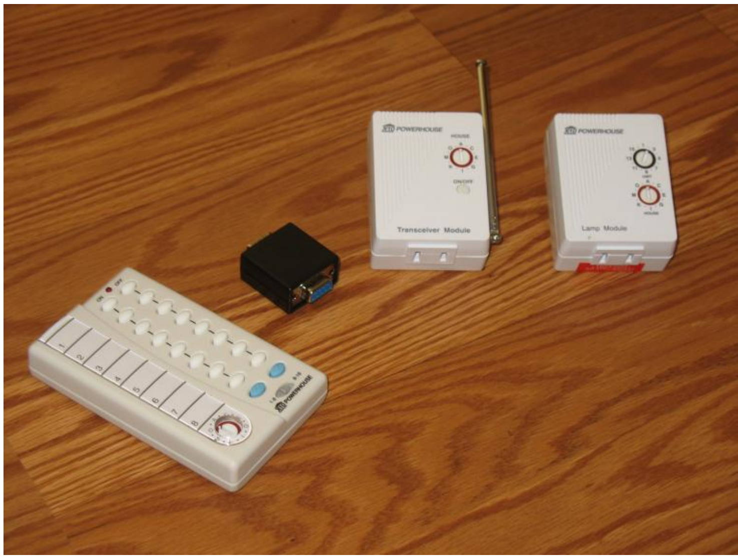


FIG. 3: X10 MODULES

As mentioned before X10 modulates using the 60Hz AC power signal that supplies all electrical sockets in the house. All the basic elements of our X10 network are shown in Figure 3.

From the right-hand side to the left-hand side, we have the Lamp Module, the Transceiver Module, the CM17A "Firecracker", and the PalmPad remote.

The transceiver module is the control system for the network. It uses its antenna to receive RF signals from the PalmPad and the CM17A. It then relays these commands to the power wiring in the house. Because of this, the transceiver does not need to be in line-of-sight with the controller—typically, it's enough to keep the two of them in the same room.

The lamp module is effectively an electronically controlled switch. It does have dimming capabilities, but these are only safe to use with incandescent lights. When the appropriate code is picked up from the power wiring, it either turns the lamp on, off, or dims/brightens it.

The CM17A is much like the PalmPad—it transmits RF signals to a Transceiver. However, you'll notice that it has no buttons on it. This is because the CM17A provides an interface to other devices over RS-232. There exists a software library for our Arduino that can use this module to send RF signals, much like the PalmPad.

THE ARDUINO: LOW-LEVEL COMMANDS

The Arduino, pictured in Figure 4, is the center of our home automation system. What we see in this picture is an Arduino Duemilanove, equipped with an Ethernet Shield, a breadboard holding the circuits to drive the IR blasters, and the CM17A, connected to the board via a solder-type 9-pin D-sub male connector.

The Arduino's main goal is to issue primitives—specific commands like TV Power, Stereo Input, Lights On, etc. It has code within it to pulse specific IR blasters with specific IR codes. Each device in our home theater system has an IR blaster taped to its remote eye.

We have the CM17A set up to communicate with a transceiver plugged into a wall socket nearby. A lamp module is connected between the floor lamp in our living room and the wall socket. (A block diagram is provided later on).

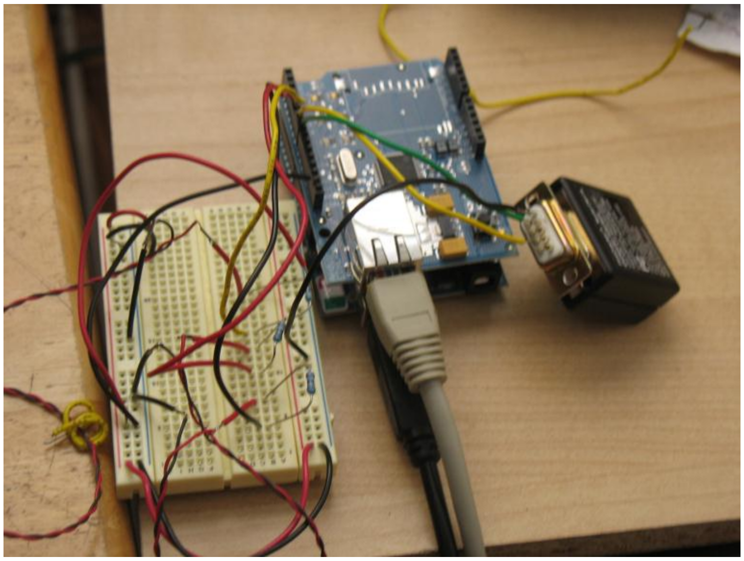


FIG. 4: OUR ARDUINO CIRCUIT

While it is possible to control the system via a telnet connection, it's not ideal. The method of input is very inconvenient, and it requires the controlling device to run a telnet client locally.

THE WEBSERVER: HIGH-LEVEL COMMANDS

As mentioned before, managing the home through a telnet connection is not desirable. We'd much rather have a single user interface which requires no knowledge of the underlying code and network architecture. Figure 5 shows such a page. This website was developed with HTML, CSS, and is accessible on any internet enabled device in the same LAN as the webserver.

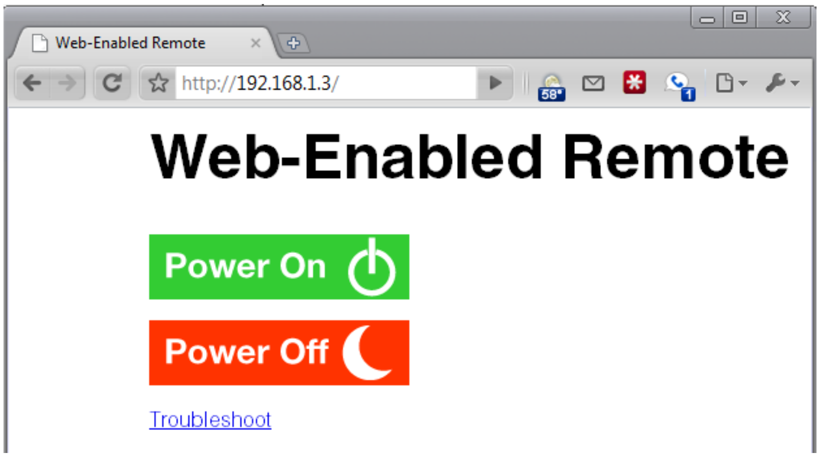


FIGURE 5: A SCREENSHOT OF THE WEBPAGE

JUSTIFICATION

In the web-based home-automation system IR blasters are used to control the Samsung Dynaflat CRT television, Samsung Home Theatre in a box and the Microsoft Xbox 360. The X10 portion consists of the X10 transceiver PL513 module used to connect to the power lines, the X10 lamp module, and the X10 firecracker to send RF commands to the X10 transceiver to control the lighting with in the house. All these functionalities are then implemented via a webserver.

IR BLASTERS

IR blasters are used to eliminate the use of remotes. People constantly misplace their remotes and it is very inconvenient to manually go to the device every time and implement certain functionality. Second, home-made IR blasters are very cheap to implement. Infra-red light emitting diodes (IR LED's) are very cheap and in order to make home-made IR blasters one only needs to solder wires onto the anode and cathode of the diode. Other methods can be used however, they are more expensive. Wireless is another possible route however, it is very expensive to implement and is more complicated to use.

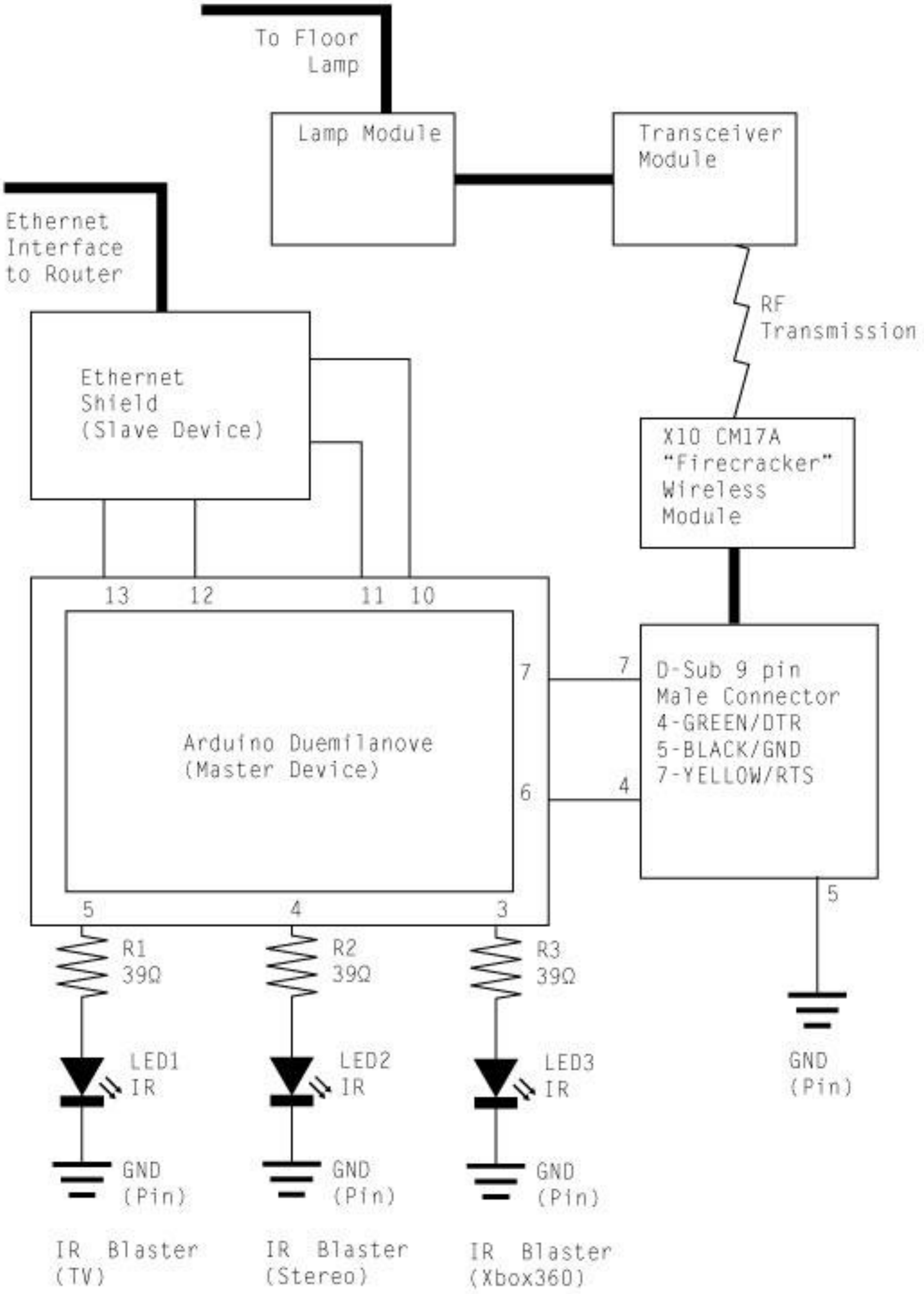
X10 SYSTEM

The X10 home automation system has been an open industry standard for remote control of lights, appliances, cameras, security systems, etc. The system works by using a mesh network of transceivers and appliance modules and functions by using power line carrier waves and RF transmissions. The power of the X10 system comes in its ability to take advantage of existing housing wire, eliminating the need for additional cables or costly installs. This makes it ideal for setup in a house or apartment.

WEB-SERVER

The easiest way to implement web-server functionality is to use the Arduino ethernet shield that plugs directly onto the Arduino board. Then run PHP scripts on a web-server on the home network. The web-server will then be able to communicate to the Arduino via ethernet connection. This way a user can go to a webpage on their network, and remotely control the Arduino, and by extension, the entire home! Also, since this is on the web multiple people can access this program and configure it to their own needs making the system expandable to other devices as well as other people.

SCHEMATIC


To Floor
Lamp
Lamp Module
Transceiver
Module
RF
Transmission
Ethernet
Interface
to Router
Ethernet
Shield
(Slave Device)
X10 CM17A
"Firecracker"
Wireless
Module
13
12
11
10
Arduino Duemilanove
(Master Device)
7
6
7
4
D-Sub 9 pin
Male Connector
4-GREEN/DTR
5-BLACK/GND
7-YELLOW/RTS
5
5
4
3
R1
39Ω
R2
39Ω
R3
39Ω
LED1
IR
LED2
IR
LED3
IR
GND
(Pin)
GND
(Pin)
GND
(Pin)
GND
(Pin)
IR Blaster
(TV)
IR Blaster
(Stereo)
IR Blaster
(Xbox360)

PARTS

Web-based home-automation system consists of:

Hardware

PARTS:	**Price:**	**Manufacturer:**
Arduino Duemilanove	$30.00	Arduino
Arduino Ethernet Shield	$35.00	Arduino
X10 lamp module	$13.00	X10
X10 transceiver module	$13.00	X10
X10 firecracker module	$7.00	X10
D-sub male connectors	$0.60	Astrex Electronics
Ethernet Cable	$1.00	UPO Technical Products
3 IR LED's	$6.00	Bright LED Electronics
Three 39 ohm resistors	$1.00	EMC Technology
TOTAL:	$106.60	

Software (all free software)

- Arduino microcontroller code based on a hybrid of Processing and C++
- Perl
- HTML
- PHP
- Telnet
- Apache
- CSS

INSTALLATION INSTRUCTIONS

PARTS USED IN INSTALL

- Arduino Duemilanove
- Arduino Ethernet Shield
- X10 lamp module

- X10 transceiver module
- X10 firecracker module
- D-sub male connectors
- Ethernet Cable
- 3 IR LED's
- Three 39 ohm resistors
- Microsoft Xbox 360
- Samsung DynaFlat CRT
- Samsung Home Theater in a Box
- Desktop running Ubuntu Linux 9.10

Note: The IR blasters were programmed for equipment listed above. For other equipment it will be necessary to write your own code.

Note: The product comes with the circuit board already assembled. One needs only to plug the appropriate pins to the Arduino

SETUP STEPS

1. Plug in Arduino Duemilanove with appropriate 5V power source adapter.
2. Load the final_build.pde file onto the Arduino Duemilanove.
3. Connect the ethernet cable from the Arduino Duemilanove to the router. DO NOT POWER ON THE ARDUINO AT THIS TIME. THE ETHERNET SHIELD HAS NOT BEEN CONFIGURED TO WORK WITH YOUR ROUTER YET.
4. On the router assign the IP address 192.168.1.90 to the MAC address DE:AD:BE:EF:FE:ED.
5. If unsure how to do step 4 consult your router's documentation.
6. Install the IR blasters matching the pin numbers on the circuit board to the pin numbers on the Arduino. For a list of connections, see Table 1.
7. Plug the X10 Transceiver into a wall socket in the same room as your home theater.
8. Connect the X10 Lamp module to the floor lamp in this room, then connect the lamp module to a wall socket.
9. Plug the X10 Firecracker into the Serial Port on the circuit board.
10. Log into the Ubuntu Desktop as a computer administrator, and run the following commands:
    a. `sudo apt-get install php5` *NOTE Accept all dependencies.*
    b. `mkdir ~/public_html/`
    c. `cd /etc/apache2/sites-available/`
    d. `sudo cp default default.old`
    e. `sudo gedit default`

        i. In this file, change “`DocumentRoot /var/www`” to “`DocumentRoot /home/<username>/public_html`”, and “`<Directory /var/www/>`” to “`<Directory /home/<username>/public_html>`”

    f. `sudo /etc/init.d/apache2 restart`

11. Once done with the setup, copy all the files from the “code” directory of this software package to your public_html
12. use ifconfig (iwconfig if your computer connects via a WiFi card) to find your MAC.
13. On your router, assign 192.168.1.3 to your computer's MAC.
14. To verify, point your browser to 192.168.1.3. You should see the frontend to the remote interface.

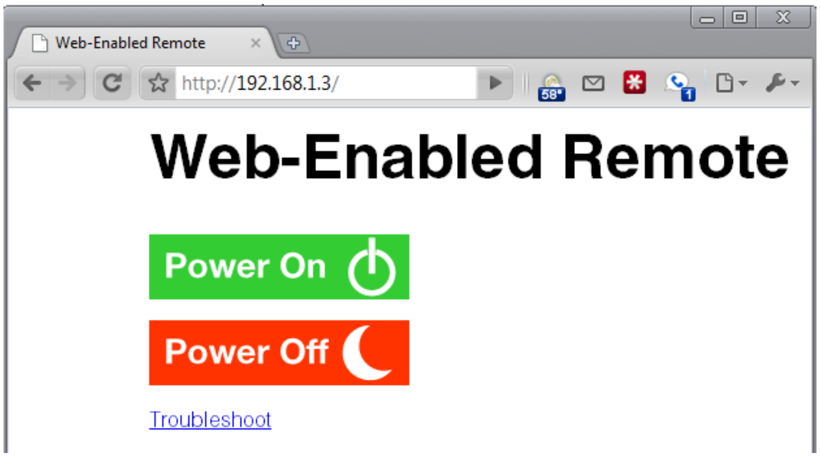
Pin	Connection
3	IR blaster for XBOX 360
4	IR blaster for Samsung Home Theater System
5	IR blaster for Samsung Dynaflat TV
6	RTS pin of X10 Firecracker
7	DTR pin of X10 Firecracker
GND	Ground pin of Arduino

TABLE 1: PINS FOR CIRCUIT BOARD

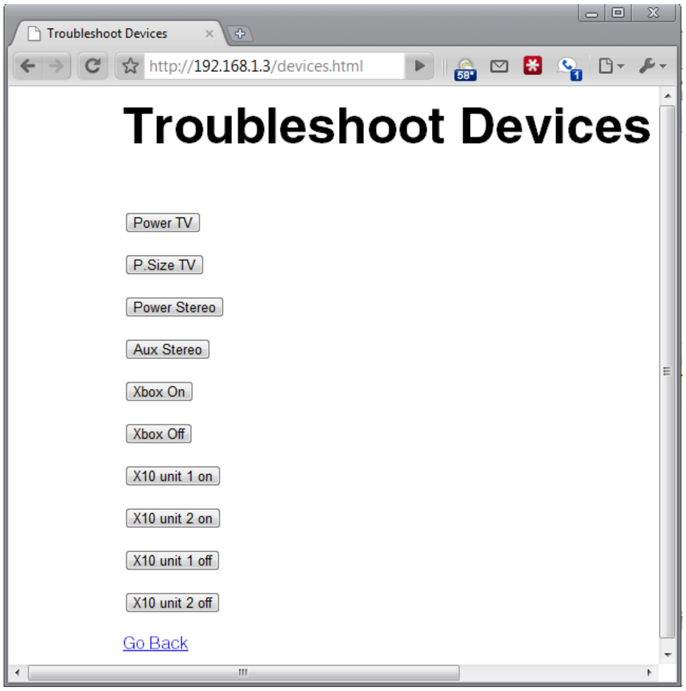
OPERATING INSTRUCTIONS

WEBSERVER

1. Type in 192.168.1.3 as the URL and you should get a webpage that looks like this.



2. If you want to turn on all the devices click Power On. If you want to turn off all devices click Power Off. If you want to just work with specific devices just click Troubleshoot.



3. Then to perform the action just click on the specific button and the action will be completed.

TELNET

1. First of all log into telnet by opening the terminal and typing the command: telnet 192.168.1.90 1023
2. If you successfully telnet you should get a screen that looks like this:

```
ansumd@ubuntu:~$ telnet 192.168.1.90 1023

Trying 192.168.1.90...

Connected to 192.168.1.90.

Escape character is '^]'.
```

3. Then type any character and press enter to display the menu:

```
k

Hey! That's not a valid input.

Telnet Remote Control: Menu

0)  TV Power

1)  TV P.Size

2)  Stereo Power

3)  Stereo Aux
```

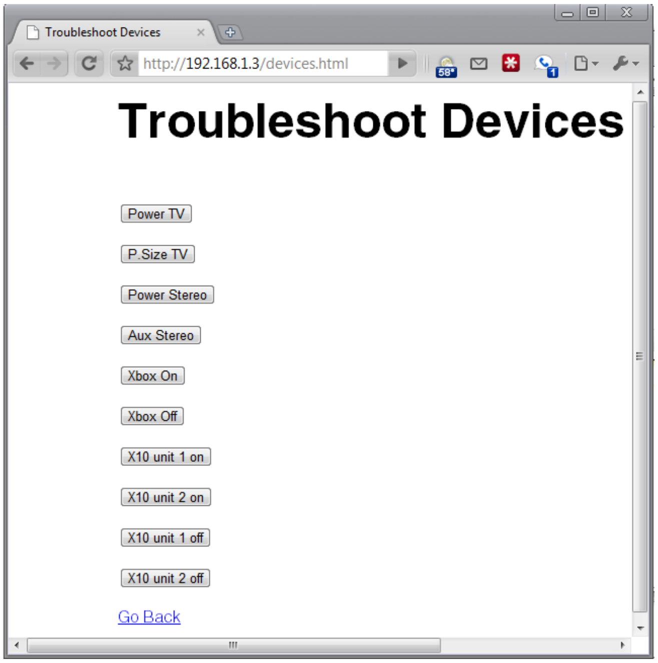
```
4) Xbox On
5) Xbox Off
6) X10 Unit 1 On
7) X10 Unit 2 On
8) X10 Unit 1 Off
9) X10 Unit 2 Off
```

4. From the menu press the appropriate number which is displayed right near the command to execute that command, and then hit enter.
5. Once you are finished press Ctrl +] to exit and then enter quit to exit telnet.

TROUBLESHOOTING

WEBSERVER

1. If the web-page is not running correctly then click the troubleshoot option on the webpage.
2. Once you hit the troubleshoot button you should get a menu that looks like this:



3. Once you get this menu click on the specific button that is creating the problem.
4. If everything seems to be okay with the webserver then you have fixed your problem, however if you still have problems then see Telnet troubleshoot.

TELNET

1. First of all log into Telnet by opening the terminal and typing the command: telnet 192.168.1.3 1023.
2. If this does not allow you to log into telnet make sure that the Arduino is powered properly and the circuitry is wired properly. Also reset the Arduino. (Note by removing the power supply or resetting the Arduino the same code will still be in the Arduino) If you successfully Telnet in you should get a screen that looks like this:

```
ansumd@ubuntu:~$ telnet 192.168.1.90 1023

Trying 192.168.1.90...

Connected to 192.168.1.90.

Escape character is '^]'
```

3. Then type any character and press enter to display the menu which should look like this:

```
k

Hey! That's not a valid input.

Telnet Remote Control: Menu

 0) TV Power

 1) TV P.Size

 2) Stereo Power

 3) Stereo Aux

 4) Xbox On

 5) Xbox Off

 6) X10 Unit 1 On

 7) X10 Unit 2 On

 8) X10 Unit 1 Off

 9) X10 Unit 2 Off
```

4. Once you are logged in then you should be able to individually check all the different options and see which ones work and which ones do not work.
5. If this still does not work then you will have to go to the raw Arduino code and run it and see if that works.
6. If the raw arduino code does not work check all hardware connections and make sure all the devices are running properly.
7. Press ctrl +] to exit and then enter quit to quit telnet.

SOFTWARE

All the software files are included in this package:

- all_off.pdf – PHP macro that turns off all the devices in the home-automation system.
- all_on.pdf – PHP macro that turns on all the devices in the home-automation system.
- final_build.pdf – Code that runs on the arduino and accepts input from telnet.
- index_html.pdf – The main web page that provides links to all_on.php, all_off.php and devices_html.php
- devices_html.php – The web page with in the main page that allows one to execute different commands associated with different devices.

ACKNOWLEDGEMENTS

None of this code would be possible without the work of Mark Ivey, and his blog post at: http://zovirl.com/2008/11/12/building-a-universal-remote-with-an-arduino/. While the code belongs to us, it is derived largely from his blog.

Thanks to Ken Shirrif at www.arcfn.com, for his work on IR libraries, and his gracious support. He owns the IRremote library, which we use for communication with the Xbox360.

Thanks to BroHogan at brohogan.blogspot.com, for his work on X10, and the CM17A "Firecracker" protocol which we use in our work. He owns the X10Firecracker library.

Thanks to the Arduino Forums and the many members of the community who have shared their code, their wisdom, and their experiences.

Feel free to use our code, but do give credit to us, as we have done for those who have supported us. ;)

May you never need your remotes again!

Best,

Ansum Dholakia

Vishal Kotcherlakota