Reflectron 2000

Instruction Manual and Laboratory Report

Nathan Decker - Spring 2017

Table of Contents

Specifications and Info

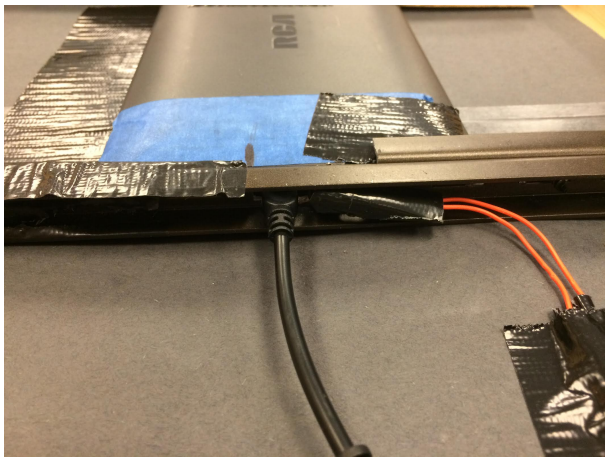
Instruction Manual

Setup

Setting Up the Reflectron 2000

1. In order to set up your mirror, you must first plug it in and let the battery charge. After an hour, you should be ready.

2. Unplug the two cables holding the tablet in place. One should be orange, and another black.



3. Slide the tablet to the left as far as possible, then pull the bottom outward to remove it.

4. Turn the tablet on, and open the settings.

5. Connect to your home WiFi network, and then open the Chrome browser.

6. Navigate to the page: https://mirrormirror.bitballoon.com

7. Place the tablet back by inserting in on the left hand side of the mirror and sliding it right. Then plug the two cords back in place.

8. At this point, you are ready to hang the mirror.

Setting Up Your Phone

1. On your phone, open your WiFi settings and connect to the network “Arduino Yun-90A2DAF70BCF”

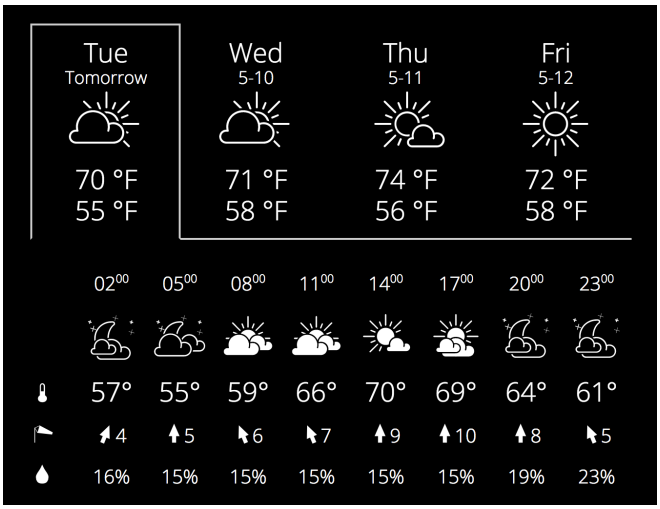
2. Navigate to the page http://192.168.240.1//sd/arduinoside.html, and pin the WebApp to your homescreen.

3. The app should appear as it does below.

Features

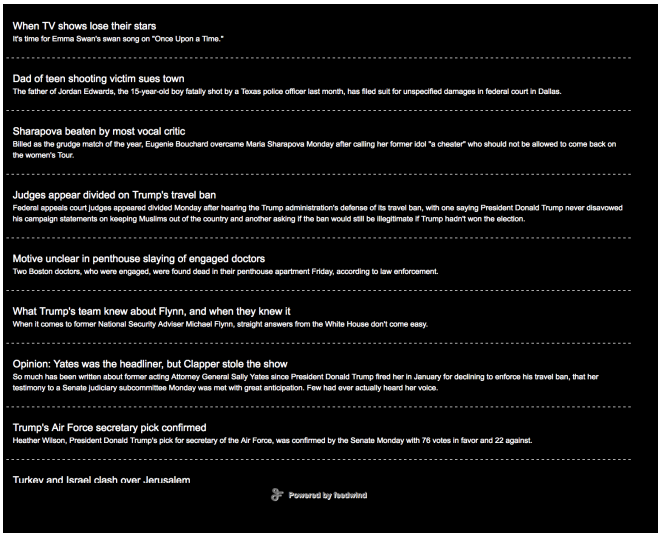
Weather

Your Reflectron 2000 is capable of giving you a weather report. Either press the "Weather" button on your phone, or say "mirror, mirror on the wall", followed by "weather" to view a five day weather forecast.



News

Your Reflectron 2000 can also give you the latest headlines. Either press the "News" button on your phone, or say "mirror, mirror on the wall", followed by "news" to view the latest headlines.



Stock Prices

Your Reflectron 2000 can also give you the latest stock quotes. Either press the "Stocks" button on your phone, or say "mirror, mirror on the wall", followed by "stocks" to view the latest stock quotes.

Light

Your mirror is equipped with a Knight Rider style light. To activate it, simply press the "Light On" button on your phone. To deactivate it, press the "Light Off" button.

Lock/Unlock

Your mirror can be locked and unlocked with the "Lock" and "Unlock" buttons on your phone. While locked, the mirror cannot act on voice commands.

Troubleshooting

Connectivity Problems

WebApp is not loading

- Your phone is most likely not connected to the correct WiFi network. Open your WiFi settings and connect to the network named "Arduino Yun-90A2DAF70BCF"

Tablet is not loading content

- Ensure that the tablet is connected to the internet by checking its WiFi settings.

Voice Recognition

Voice recognition is working sporadically

- Try speaking loudly and over annunciating. Turning your face towards the mirror can also help.

Voice recognition is not working at all

- Try reloading the web page on the tablet.
- Check that the WiFi on the tablet is connected

Power

The tablet is not turning on

- The lights, WiFi, and tablet itself are all powered by the same battery (inside the tablet). In order for these features to work together, you must ensure that the battery is charged before starting the mirror.

Laboratory Report

Diagrams for Overview

Figure 1: Diagram of System Setup

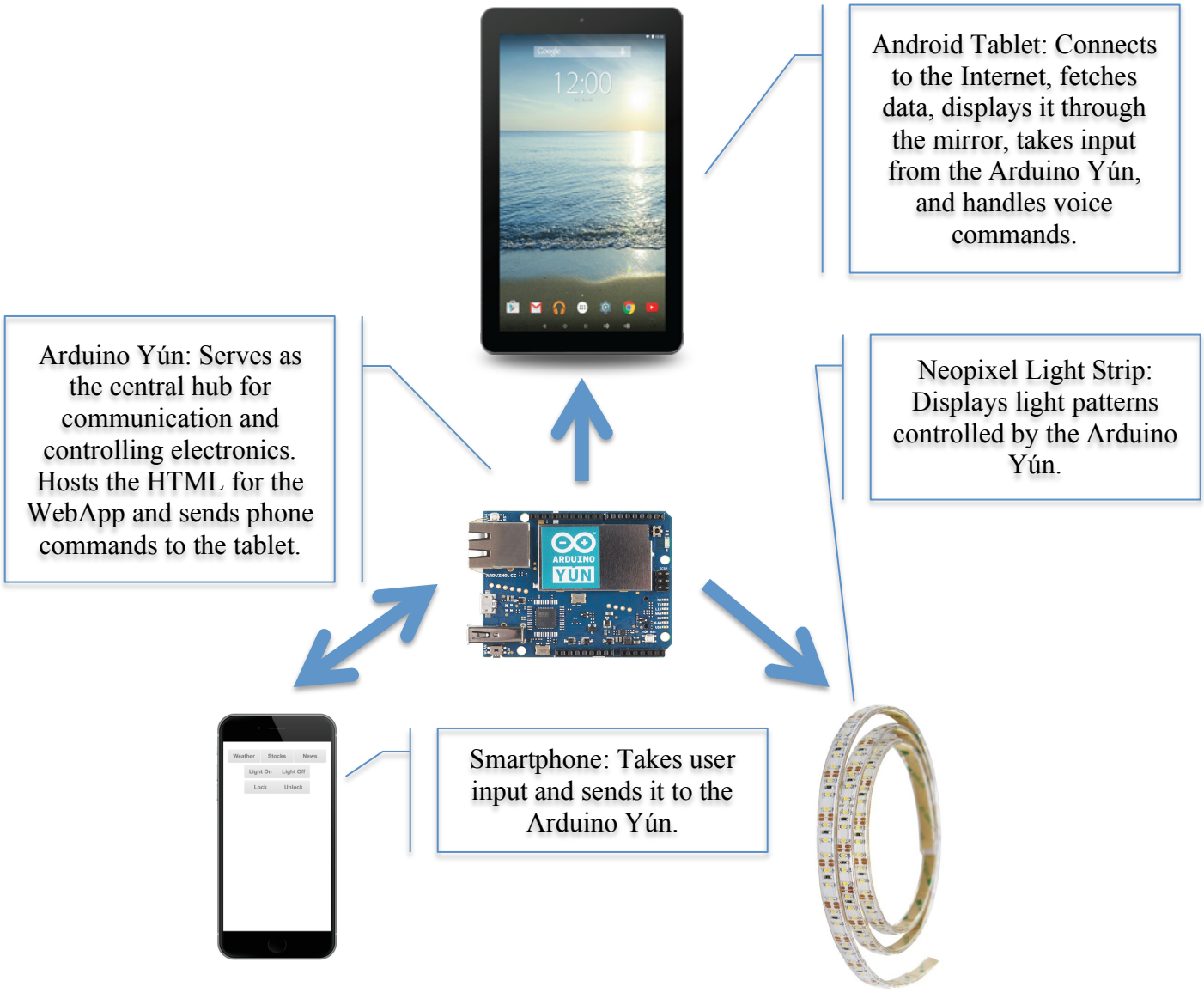
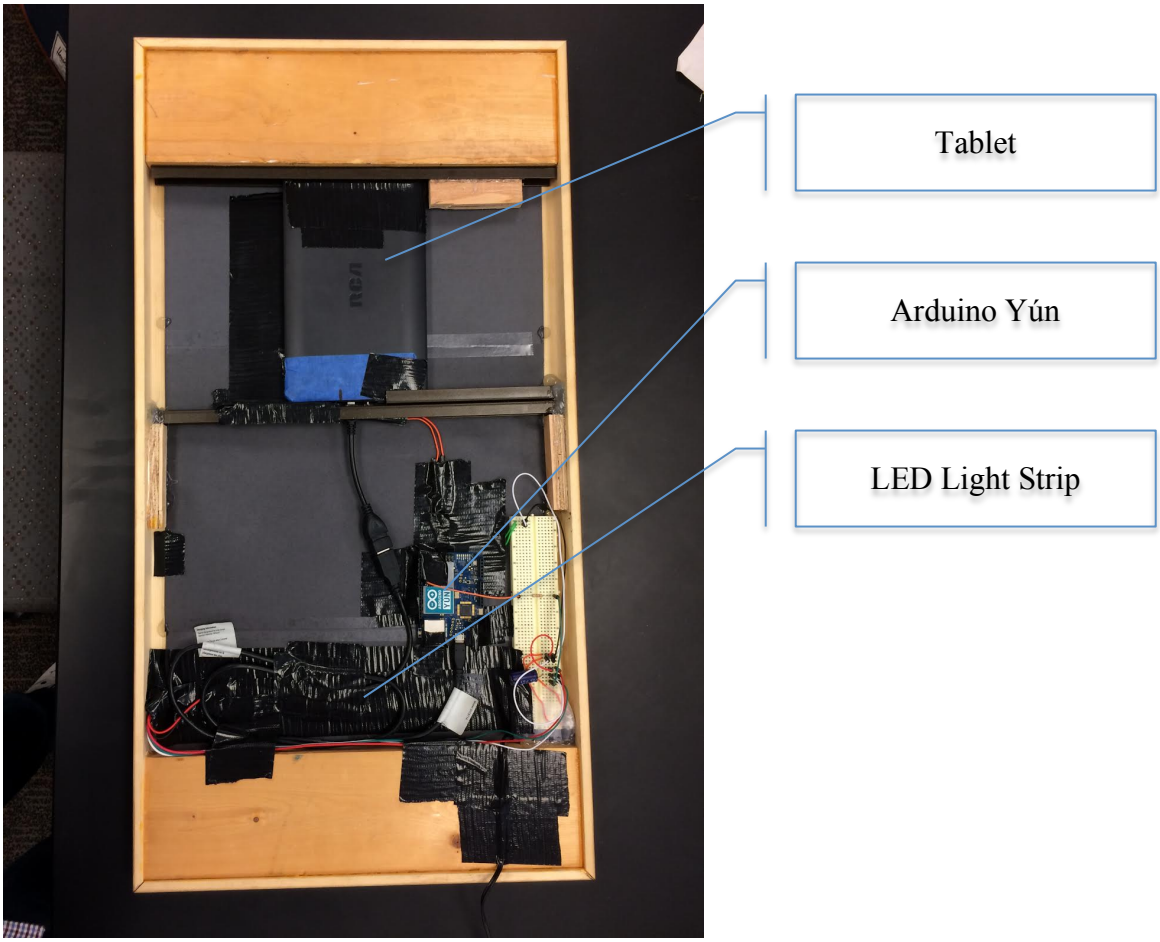


Figure 2: Photo of Setup



Code Samples

Arduino Code

```
#include <Bridge.h>
#include <BridgeServer.h>
#include <BridgeClient.h>
#include <Keyboard.h>
int lightcheck = 0;
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
  #include <avr/power.h>
#endif
#define LIGHTCYCLES 1
#define LIGHTSPEED 50
#define LIGHTWIDTH 2
#define LIGHTCOLOR 0xFF1000
#define PIN 6
#define NUM_PIXELS  8
Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUM_PIXELS, PIN, NEO_GRB + NEO_KHZ800);

BridgeServer server; // Declares server

void setup() {

  Bridge.begin(); // Begins bridge to connect Linux processor to Arduino
  server.listenOnLocalhost(); // Listens for requests
  server.begin();
  strip.begin();
  clearStrip(); // Turns all pixels off

}

void loop() {

  BridgeClient client = server.accept();   // Get clients coming from server

  if (client) {
    process(client);
    client.stop();
  }

  delay(50);


  if(lightcheck==1){ // If light should be going
    knightRider(LIGHTCYCLES, LIGHTSPEED, LIGHTWIDTH, LIGHTCOLOR); // Turns on light
    clearStrip();
  }
  else {
    clearStrip(); // Turn strip off
  }

}

// Function to handle various commands sent by phone

void process(BridgeClient client) { // If command is sent by phone

  String command = client.readStringUntil('/'); // Reads the command

  if (command == "weather") {
    Keyboard.begin();
    Keyboard.write('w'); // Sends a W to the tablet
    Keyboard.end();
  }
```

```
  if (command == "unlock") {
    Keyboard.begin();
    Keyboard.write('t'); // Sends a T to the tablet
    Keyboard.end();
  }

  if (command == "stocks") {
    Keyboard.begin();
    Keyboard.write('s'); // Sends an S to the tablet
    Keyboard.end();
  }

  if (command == "news") {
    Keyboard.begin();
    Keyboard.write('n'); // Sends an N to the tablet
    Keyboard.end();
  }

  if (command == "lock") {
    Keyboard.begin();
    Keyboard.write('l'); // Sends an l to the tablet
    Keyboard.end();
  }

  if (command == "lighton") {
    lightcheck = 1;
  }

  if (command == "lightoff") {
    lightcheck = 0;
  }
}

// Function to handle overall light effect written by Brett Walach
// https://github.com/technobly/NeoPixel-KnightRider
// Used under the MIT License

/* Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE. */

void knightRider(uint16_t cycles, uint16_t speed, uint8_t width, uint32_t color) {
  uint32_t old_val[NUM_PIXELS];
  for(int i = 0; i < cycles; i++){
    for (int count = 1; count<NUM_PIXELS; count++) {
      strip.setPixelColor(count, color);
      old_val[count] = color;
      for(int x = count; x>0; x--) {
        old_val[x-1] = dimColor(old_val[x-1], width);
        strip.setPixelColor(x-1, old_val[x-1]);
      }
      strip.show();
      delay(speed);
    }
    for (int count = NUM_PIXELS-1; count>=0; count--) {
```

```
      strip.setPixelColor(count, color);
      old_val[count] = color;
      for(int x = count; x<=NUM_PIXELS ;x++) {
        old_val[x-1] = dimColor(old_val[x-1], width);
        strip.setPixelColor(x+1, old_val[x+1]);
      }
      strip.show();
      delay(speed);
    }
  }
}

// Function to clear the strip of all colors

void clearStrip() {
  for( int i = 0; i<NUM_PIXELS; i++){
    strip.setPixelColor(i, 0x000000); strip.show();
  }
}

// Function for the dim effect

uint32_t dimColor(uint32_t color, uint8_t width) {
   return (((color&0xFF0000)/width)&0xFF0000) + (((color&0x00FF00)/width)&0x00FF00) +
(((color&0x0000FF)/width)&0x0000FF);
}
```

Sample Internet Deployed HTML/Javascript

```
<!DOCTYPE html>
<html>
<head>
  <title>Triggered</title>

  <style>
  body {
  background-color: black;
  img.center {
    display: block;
    margin: 0 auto;
  }
  }
  </style>

<script>

    // This function works to listen for keyboard inputs given by the Arduino Yun
    // Each input will open a different page

    window.addEventListener("keydown", doSomething, false);

    function doSomething(e) {
    switch(e.keyCode) {
        case 76:
            window.open ('https://mirrormirror.bitballoon.com/lockscreen','_self',false)
            break;
        case 84:
            window.open ('https://mirrormirror.bitballoon.com/','_self',false)
            break;
        case 83:
            window.open ('https://mirrormirror.bitballoon.com/stock','_self',false)
            break;
        case 87:
            window.open ('https://mirrormirror.bitballoon.com/weather','_self',false)
            break;
        case 78:
            window.open ('https://mirrormirror.bitballoon.com/news','_self',false)
```

```
        break;
    }
}

</script>

</head>

<body>

<img src="triggered.gif" style="width:800px;height:900px;">

  <script src="annyang.min.js"></script>
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
  <script>
  "use strict";

   function loadUrl(newLocation)
  {
  window.location = newLocation;
  return false;
  }

  if (annyang) {

    // Defines the functions our commands will run
    var stock = function() {
      return loadUrl('https://mirrormirror.bitballoon.com/stock');
    };

    var weather = function() {
      return loadUrl('https://mirrormirror.bitballoon.com/weather');
    };

    var fairest = function() {
      return loadUrl('https://mirrormirror.bitballoon.com/fairest');
    };

    var news = function() {
      return loadUrl('https://mirrormirror.bitballoon.com/news');
    };

    // Defines the possible voice commands
    var commands = {
      'stock': stock,
      'who\'s the fairest of them all': fairest,
      'news': news,
      'weather': weather,
    };

    // Add voice commands to respond to
    annyang.addCommands(commands);

    // Sets default language
    annyang.setLanguage('en');

    // Starts sending every word you say to Google...  Totally fine.
    annyang.start();
  } else {
    $(document).ready(function() {
      $('#unsupported').fadeIn('fast');
    });
  }

  </script>

</body>

</html>
```

Sample Arduino Yún Deployed HTML/CSS for WebApp

```
<!DOCTYPE html>

<html>
<head>
<meta name="apple-mobile-web-app-capable" content="yes" />
<title>Nathan's Smart Mirror</title>
<style type="text/css">
.classname {
	-moz-box-shadow:inset 0px 1px 0px 0px #ffffff;
	-webkit-box-shadow:inset 0px 1px 0px 0px #ffffff;
	box-shadow:inset 0px 1px 0px 0px #ffffff;
	background:-webkit-gradient( linear, left top, left bottom, color-stop(0.05, #ededed), color-stop(1, #dfdfdf) );
	background:-moz-linear-gradient( center top, #ededed 5%, #dfdfdf 100% );
	filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#ededed', endColorstr='#dfdfdf');
	background-color:#ededed;
	-webkit-border-top-left-radius:6px;
	-moz-border-radius-topleft:6px;
	border-top-left-radius:6px;
	-webkit-border-top-right-radius:6px;
	-moz-border-radius-topright:6px;
	border-top-right-radius:6px;
	-webkit-border-bottom-right-radius:6px;
	-moz-border-radius-bottomright:6px;
	border-bottom-right-radius:6px;
	-webkit-border-bottom-left-radius:6px;
	-moz-border-radius-bottomleft:6px;
	border-bottom-left-radius:6px;
text-indent:0;
	border:1px solid #dcdcdc;
	display:inline-block;
	color:#777777;
	font-family:arial;
	font-size:43px;
	font-weight:bold;
	font-style:normal;
height:100px;
	line-height:100px;
width:250px;
	text-decoration:none;
	text-align:center;
	text-shadow:1px 1px 0px #ffffff;
}.classname:hover {
	background:-webkit-gradient( linear, left top, left bottom, color-stop(0.05, #dfdfdf), color-stop(1, #ededed) );
	background:-moz-linear-gradient( center top, #dfdfdf 5%, #ededed 100% );
	filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#dfdfdf', endColorstr='#ededed');
	background-color:#dfdfdf;
}.classname:active {
	position:relative;
	top:1px;
}</style>
</head>
<body>
<div style="text-align:center;">
</br>
<a href="http://192.168.240.1/sd/weather.html" target="_self" class="classname">Weather</a>
<a href="http://192.168.240.1/sd/stocks.html" target="_self" class="classname">Stocks</a>
<a href="http://192.168.240.1/sd/news.html" target="_self" class="classname">News</a></br></br>
<a href="http://192.168.240.1/sd/lighton.html" target="_self" class="classname">Light On</a>
<a href="http://192.168.240.1/sd/lightoff.html" target="_self" class="classname">Light Off</a></br></br>
<a href="http://192.168.240.1/sd/lock.html" target="_self" class="classname">Lock</a>
<a href="http://192.168.240.1/sd/unlock.html" target="_self" class="classname">Unlock</a>
</div>

<IFRAME width=1 height=1 src=http://192.168.240.1/arduino/weather/42 scrolling=no frameborder=0></IFRAME>
</body>
</html>
```

Equipment Specifications

Component	Model/Serial Number
Tablet	RCA Voyager 7" 16GB Tablet RCT6873W42 CHAR
Arduino Board	Arduino Yún DEV-12053
LED Light Strip	Mokungit 3.2FT 1M WS2812B
USB Cables	n/a
Power Source	ZOZO 12W Power Adapter
Acrylic Sheet	12" x 24" Acrylic See-Through Mirror B01G4MQ3WQ
Breadboard	n/a
Miscellaneous Hookup Wire	n/a

List of API Dependencies

- The voice recognition depends on a program called Annyang, which uses the Google Speech Recognition API

  Annyang: https://github.com/TalAter/annyang
  Google Speech API: https://cloud.google.com/speech/

- The weather forecast depends on an API from Meteoblue

  MeteoBlue: https://www.meteoblue.com

- The stock market data comes from a TradingView API

  TradingView: https://www.tradingview.com/widget/

- The news data comes from CNN's RSS feed

  CNN: http://rss.cnn.com/rss/cnn_topstories.rss

General Notes

- It should be noted that this mirror warrants privacy concerns since it sends every word you say in its presence to Google. As far as I can tell, your words are not tagged with your identity, however there are no guarantees in this regard. Most hardware gets around this concern by using a local speech recognition program to listen for a key word, and will only send data to a server after this word has been recognized. This system is not implemented in this current version of the mirror.

- It might be possible to incorporate an external microphone to improve the accuracy/ease of speech recognition. I do, however like the fact that the microphone isn't too powerful due to the above security concerns.