Byrne Seminar

Internet of Things Things

kulikows@cs.rutgers.edu
m cgrew@cs.rutgers.edu
Week 1: Definitions, mapping out the course
Week 2: Composition of an IOT Thing. Programming it. The IOT before the “I”: Apollo AGC (1961-71)
Week 3: Introduction to our processor: The Arduino Distribution of hardware **WHICH I WANT BACK**
Week 4: Security of IOT things. Fiddling with devices
Week 5: Getting used to the Arduino and building devices
Week 6: Show-offs; turn in your code Uses of IOTs: medicine, "smart cities", political oppression
Week 7: Research description: you pick an IOT, and research it for presentation to the class
Week 8: Security of IOT things #2: security, and liability
Week 9: Presentations of research projects part 1
Week 10: Presentations of research projects part 2
**Arduino software install**

[https://www.arduino.cc/en/Main/Software](https://www.arduino.cc/en/Main/Software)

… select and install for whatever OS you’ve got.

**Power:** either power cable OR USB cable.

**Program:** runs most recently downloaded program… Forever

Check out

A Small Electronics Primer

Electronics in Brief
No previous electronic experience is required to have fun with this kit. Here are a few details about each component to make identifying, and perhaps understanding them, a bit easier. If at any point you are worried about how a component is used or why it’s not working the internet offers a treasure trove of advice, or we can be contacted at help@oomlout.com

Component Details

**LED (Light Emitting Diode)**

**What it Does:** Emits light when a small current is passed through it. (only in one direction)

**Identifying:** Looks like a mini light bulb.

**No. of Leads:** 2 (one longer, this one connects to positive)

**Things to watch out for:**
- Will only work in one direction
- Requires a current limiting resistor

**More Details:** http://ardx.org/LED

**Diode**

**What it Does:** The electronic equivalent of a one way valve. Allowing current to flow in one direction but not the other.

**Identifying:** Usually a cylinder with wires extending from either end. (and an off center line indicating polarity)

**No. of Leads:** 2

**Things to watch out for:**
- Will only work in one direction (current will flow if end with the line is connected to ground)

**More Details:** http://ardx.org/DIOD

**Resistors**

**What it Does:** Restricts the amount of current that can flow through a circuit.

**Identifying:** Cylinder with wires extending from either end. The value is displayed using a color coding system (for details see next page)

**No. of Leads:** 2

**Things to watch out for:**
- Easy to grab the wrong value (double check the colors before using)

**More Details:** http://ardx.org/RESI

**Transistor**

**What it Does:** Uses a small current to switch or amplify a much larger current.

**Identifying:** Comes in many different packages but you can read the part number off the package. (ON2222AG in this kit and find a datasheet online)

**No. of Leads:** 3 (Base, Collector, Emitter)

**Things to watch out for:**
- Plugging in the right way round (also a current limiting resistor is often needed on the base pin)

**More Details:** http://ardx.org/TRAN

**Hobby Servo**

**What it Does:** Takes a timed pulse and converts it into an angular position of the output shaft.

**Identifying:** A plastic box with 3 wires coming out one side and a shaft with a plastic horn out the top.

**No. of Leads:** 3

**Things to watch out for:**
- The plug is not polarized so make sure it is plugged in the right way.

**More Details:** http://ardx.org/SERV

**DC Motor**

**What it Does:** Spins when a current is passed through it.

**Identifying:** This one is easy, it looks like a motor. Usually a cylinder with a shaft coming out of one end.

**No. of Leads:** 2

**Things to watch out for:**
- Using a transistor or relay that is rated for the size of motor you’re using.

**More Details:** http://ardx.org/MOTO
### Component Details (cont.)

#### Piezo Element
**What it Does:**
A pulse of current will cause it to click. A stream of pulses will cause it to emit a tone.

**Identifying:**
In this kit it comes in a little black barrel, but sometimes they are just a gold disc.

**No. of Leads:**
2

**Things to watch out for:**
- Difficult to misuse.

**More Details:**
http://ardx.org/PIEZ

#### IC (Integrated Circuit)
**What it Does:**
Packages any range of complicated electronics inside an easy to use package.

**Identifying:**
The part ID is written on the outside of the package. (this sometimes requires a lot of light or a magnifying glass to read).

**No. of Leads:**
2 - 100s (in this kit there is one with 3 (TMP36) and one with 16 (74HC595)

**Things to watch out for:**
- Proper orientation. (look for marks showing pin 1)

**More Details:**
http://ardx.org/ICIC

#### Pushbutton
**What it Does:**
Completes a circuit when it is pressed.

**Identifying:**
A little square with leads out the bottom and a button on the top.

**No. of Leads:**
4

**Things to watch out for:**
- These are almost square so can be inserted 90 degrees off angle.

**More Details:**
http://ardx.org/BUTT

#### Potentiometer
**What it Does:**
Produces a variable resistance dependant on the angular position of the shaft.

**Identifying:**
They can be packaged in many different form factors, look for a dial to identify.

**No. of Leads:**
3

**Things to watch out for:**
- Accidentally buying logarithmic scale.

**More Details:**
http://ardx.org/POTE

#### Photo Resistor
**What it Does:**
Produces a variable resistance dependant on the amount of incident light.

**Identifying:**
Usually a little disk with a clear top and a curvy line underneath.

**No. of Leads:**
2

**Things to watch out for:**
- Remember it needs to be in a voltage divider before it provides a useful input.

**More Details:**
http://ardx.org/PHOT

---

### Resistor Color Code

<table>
<thead>
<tr>
<th>Color</th>
<th>Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - Black</td>
<td>B</td>
<td>-80% - none</td>
</tr>
<tr>
<td>1 - Brown</td>
<td>B</td>
<td>-80% - none</td>
</tr>
<tr>
<td>2 - Red</td>
<td>R</td>
<td>-80% - none</td>
</tr>
<tr>
<td>3 - Orange</td>
<td>O</td>
<td>-80% - none</td>
</tr>
<tr>
<td>4 - Yellow</td>
<td>Y</td>
<td>-80% - none</td>
</tr>
<tr>
<td>5 - Green</td>
<td>G</td>
<td>-20% - none</td>
</tr>
<tr>
<td>6 - Blue</td>
<td>B</td>
<td>-10% - silver</td>
</tr>
<tr>
<td>7 - Purple</td>
<td>P</td>
<td>-5% - gold</td>
</tr>
<tr>
<td>8 - Grey</td>
<td>G</td>
<td>- none</td>
</tr>
<tr>
<td>9 - White</td>
<td>W</td>
<td>- none</td>
</tr>
</tbody>
</table>

**Examples:**
- green-blue-brown - 560 ohms
- red-red-red - 2 200 ohms (2.2k)
- brown-black-orange - 10 000 ohms (10k)

---

### Lead Clipping

Some components in this kit come with very long wire leads. To make them more compatible with a breadboard a couple of changes are required.

**LEDs:**
Clip the leads so the long lead is ~10mm (3/8") long and the short one is ~7mm (9/32").

**Resistors:**
Bend the leads down so they are 90 degrees to the cylinder. Then snip them so they are ~6mm (~1/4") long.

**Other Components:**
Other components may need clipping. Use your discretion when doing so.
LED

**Schematic**
- Arduino
  - pin 13
- LED (light emitting diode)
- longer lead
- resistor (330 ohm)
  - (orange-orange-brown)
- gnd (ground) (-)

RBG LED

**Schematic**
- Arduino
  - pin 11
  - pin 10
  - pin 9
- resistor (330 ohm)
- blue
- green
- red
- longest lead
- flat side
- gnd (ground) (-)
- common (gnd)
Ultrasonic
/* HC-SR04 Sensor


This sketch reads a HC-SR04 ultrasonic rangefinder and returns the distance to the closest object in range. To do this, it sends a pulse to the sensor to initiate a reading, then listens for a pulse to return. The length of the returning pulse is proportional to the distance of the object from the sensor.

The circuit:

- VCC connection of the sensor attached to +5V
- GND connection of the sensor attached to ground
- TRIG connection of the sensor attached to digital pin 2
- ECHO connection of the sensor attached to digital pin 4

Original code for Ping)) example was created by David A. Mellis
Adapted for HC-SR04 by Tautvidas Sipavicius

This example code is in the public domain.
*/

const int trigPin = 2;
const int echoPin = 4;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}
void loop()
{
   // establish variables for duration of the ping,
   // and the distance result in inches and centimeters:
   long duration, inches, cm;

   // The sensor is triggered by a HIGH pulse of 10 or more microseconds.
   // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
   pinMode(trigPin, OUTPUT);
   digitalWrite(trigPin, LOW);
   delayMicroseconds(2);
   digitalWrite(trigPin, HIGH);
   delayMicroseconds(10);
   digitalWrite(trigPin, LOW);

   // Read the signal from the sensor: a HIGH pulse whose
   // duration is the time (in microseconds) from the sending
   // of the ping to the reception of its echo off of an object.
   pinMode(echoPin, INPUT);
   duration = pulseIn(echoPin, HIGH);

   // convert the time into a distance
   inches = microsecondsToInches(duration);
   cm = microsecondsToCentimeters(duration);

   Serial.print(inches);
   Serial.print("in, ");
   Serial.print(cm);
   Serial.print("cm");
   Serial.println();
   delay(100);
}
long microsecondsToInches(long microseconds)
{
    // According to Parallax's datasheet for the PING)), there are
    // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
    // second). This gives the distance travelled by the ping, outbound
    // and return, so we divide by 2 to get the distance of the obstacle.
    return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds)
{
    // The speed of sound is 340 m/s or 29 microseconds per centimeter.
    // The ping travels out and back, so to find the distance of the
    // object we take half of the distance travelled.
    return microseconds / 29 / 2;
}
Temperature

![Schematic of temperature sensor](image)
Buzzer

Schematic

Arduino
pin 9

Piezo
Element

gnd
(ground) (-)
Buzzer Code

```c
/* Melody */
*(cleft) 2005 D. Cuartielles for K3
This example uses a piezo speaker to play melodies. It sends
a square wave of the appropriate frequency to the piezo,
generating the corresponding tone.
The calculation of the tones is made following the
mathematical operation:
    timeHigh = period / 2 = 1 / (2 * toneFrequency)
where the different tones are described as in the table:

<table>
<thead>
<tr>
<th>note</th>
<th>frequency</th>
<th>period</th>
<th>timeHigh</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>261 Hz</td>
<td>3830</td>
<td>1915</td>
</tr>
<tr>
<td>d</td>
<td>294 Hz</td>
<td>3400</td>
<td>1700</td>
</tr>
<tr>
<td>e</td>
<td>329 Hz</td>
<td>3038</td>
<td>1519</td>
</tr>
<tr>
<td>f</td>
<td>349 Hz</td>
<td>2864</td>
<td>1432</td>
</tr>
<tr>
<td>g</td>
<td>392 Hz</td>
<td>2550</td>
<td>1275</td>
</tr>
<tr>
<td>a</td>
<td>440 Hz</td>
<td>2272</td>
<td>1136</td>
</tr>
<tr>
<td>b</td>
<td>493 Hz</td>
<td>2028</td>
<td>1014</td>
</tr>
<tr>
<td>c</td>
<td>523 Hz</td>
<td>1912</td>
<td>956</td>
</tr>
</tbody>
</table>

http://www.arduino.cc/en/Tutorial/Melody
*/

int speakerPin = 9;
int length = 15; // the number of notes
char notes[] = "ccgaagffeedc"; // a space represents a rest
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 4 }; int tempo = 300;

void playNote(char note, int duration) {
    char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'c' };
    int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };
// play the tone corresponding to the note name
    for (int i = 0; i < 8; i++) {
        if (names[i] == note) {
            playTone(tones[i], duration);
        }
    }
}

void playTone(int tone, int duration) {
    for (long i = 0; i < duration * 1000L; i += tone * 2) {
        digitalWrite(speakerPin, HIGH);
        delayMicroseconds(tone);
    }
}

void setup() {
    pinMode(speakerPin, OUTPUT);
}

void loop() {
    for (int i = 0; i < length; i++) {
        if (notes[i] == ' ') {
            delay(beats[i] * tempo); // rest
        } else {
            playNote(notes[i], beats[i] * tempo);
        }
        // pause between notes
delay(tempo / 2);
    }
}
```