Smart Measuring Cup

Created by Tony DiCola

https://learn.adafruit.com/smart-measuring-cup

Last updated on 2023-08-29 02:31:32 PM EDT

Table of Contents

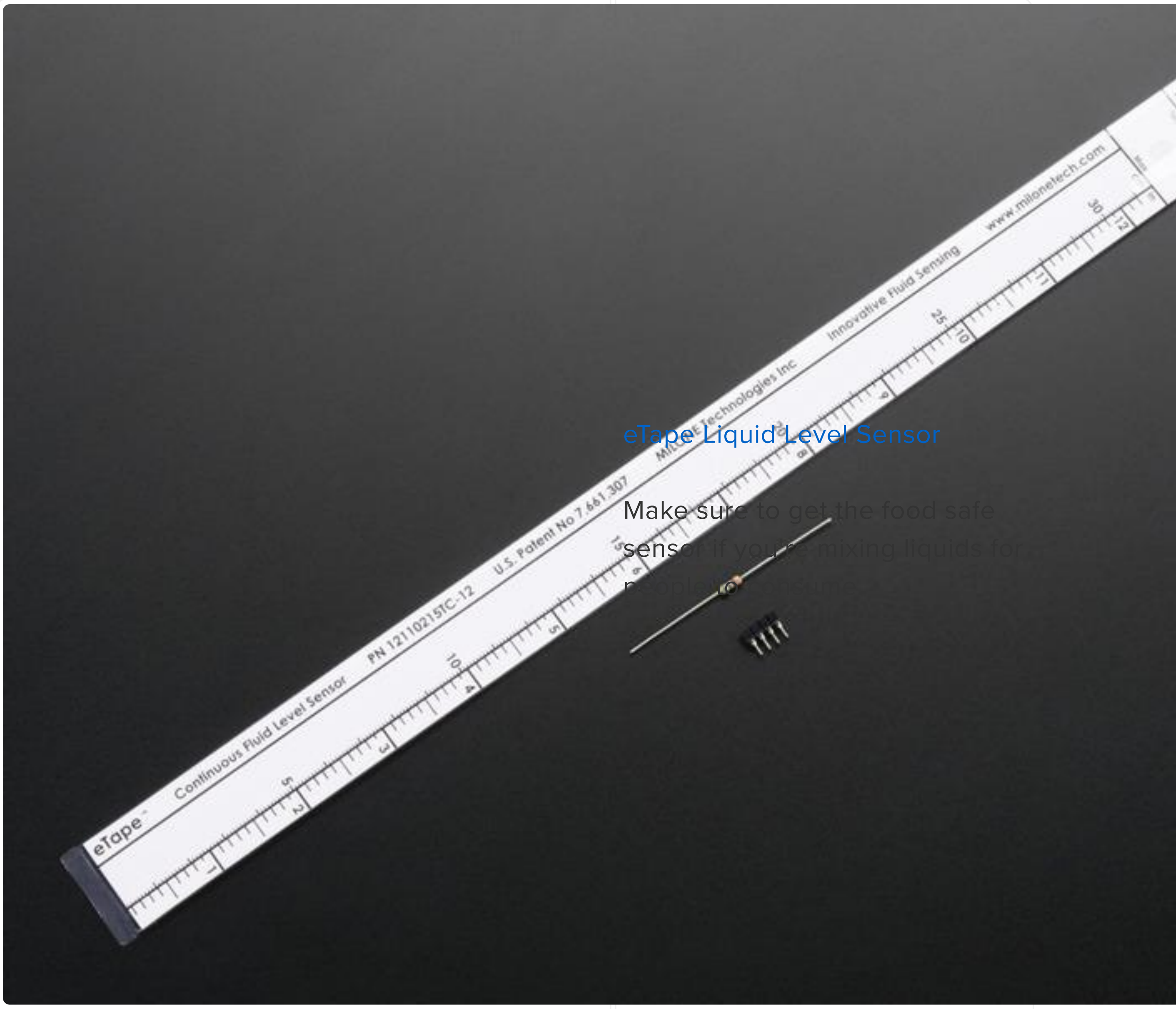
Overview

This guide will show you how to build a smart measuring cup which displays the volume of liquid in real time on a web page. No longer will you need to squint and guess while measuring liquids; the smart measuring cup's web page clearly shows the measured volume, and even converts between units or tares measurements like a digital scale!

The brains behind the smart measuring cup are an Arduino Yun () and an eTape liquid level sensor (http://adafru.it/1786). The Yun is the perfect platform for this project because it can read sensor data with it's ATmega microcontroller, and serve a web application (written in python and Flask ()) using it's Linux-based processor. By following this guide you can even learn how to send sensor data to a web page for your own projects.

Hardware

You will need the following parts for this project:



eTape Liquid Level Sensor

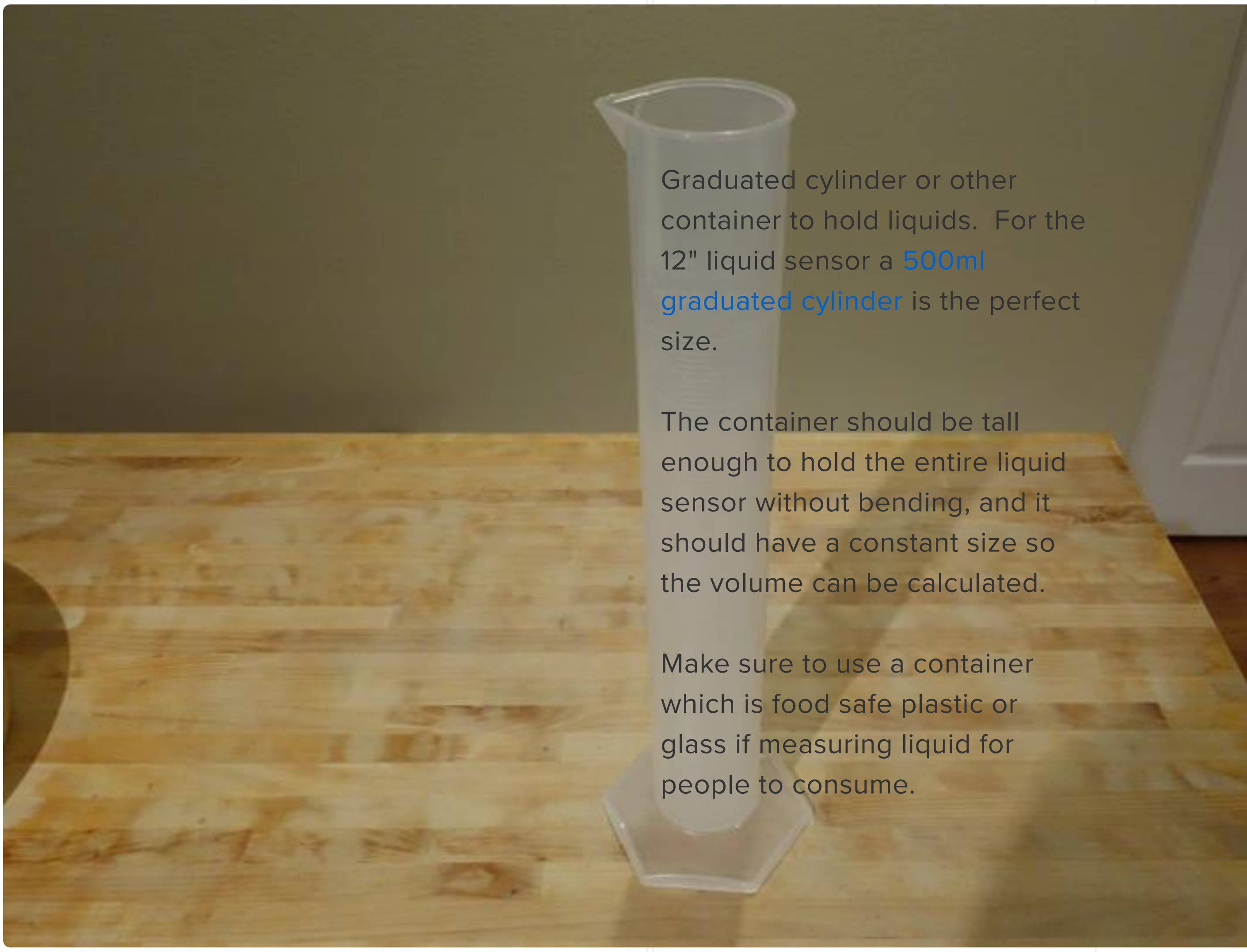
Make sure to get the food safe sensor if you're mixing liquids for people to consume.

Arduino Yun, Arduino Uno, or other microcontroller with an analog input.

If you use the Yun you can make a web page which displays the measured liquid volume.

If you use the Uno or other microcontroller you'll want to add a display, like one of these OLED graphic displays.

Female to male hookup wires. These will connect the sensor to the Arduino.



Graduated cylinder or other container to hold liquids. For the 12" liquid sensor a 500ml graduated cylinder is the perfect size.

The container should be tall enough to hold the entire liquid sensor without bending, and it should have a constant size so the volume can be calculated.

Make sure to use a container which is food safe plastic or glass if measuring liquid for people to consume.

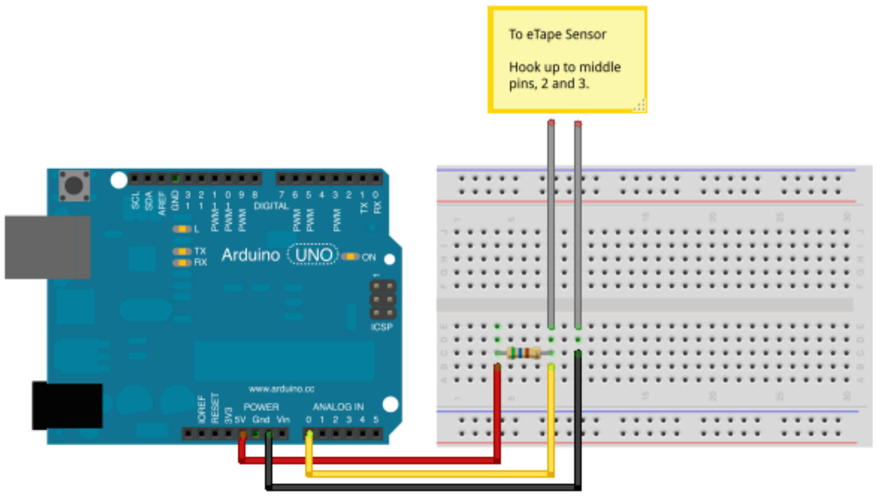
MicroSD card for storing scripts and data on the Arduino Yun. Any card with about 10-20 megabytes of free space should work.

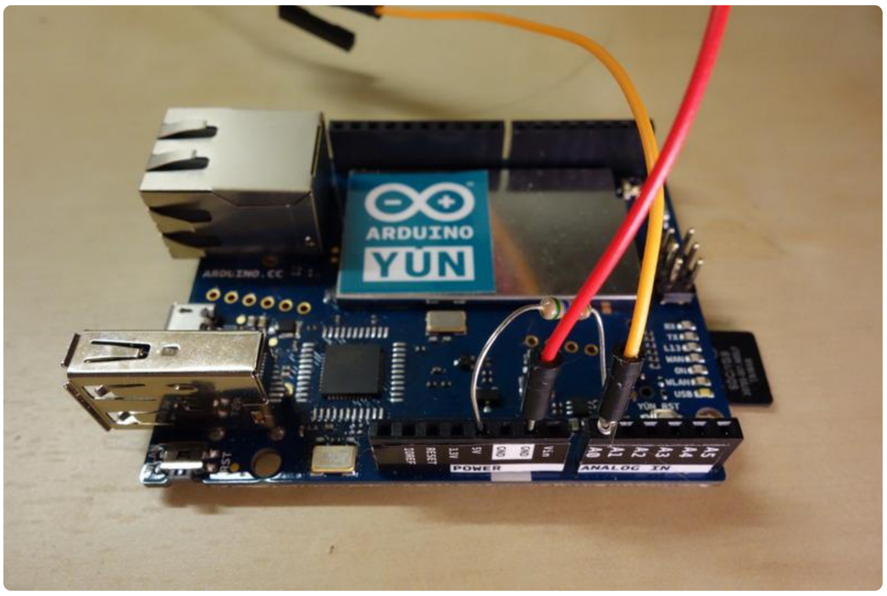
Assembly

The assembly of the hardware is very simple because the eTape liquid sensor is a resistive sensor, just like a photocell (). Attach one of the middle two leads to ground on the Arduino. Attach the other middle lead to an analog input such as analog 0 on the Arduino. Finally attach the 560 ohm resistor that comes with the sensor between the analog 0 and 5 volt pins on the Arduino.

The outer two pins on the sensor will be unused (they're for temperature compensation in a bridge configuration).

You can see a diagram and photo of the wiring below:

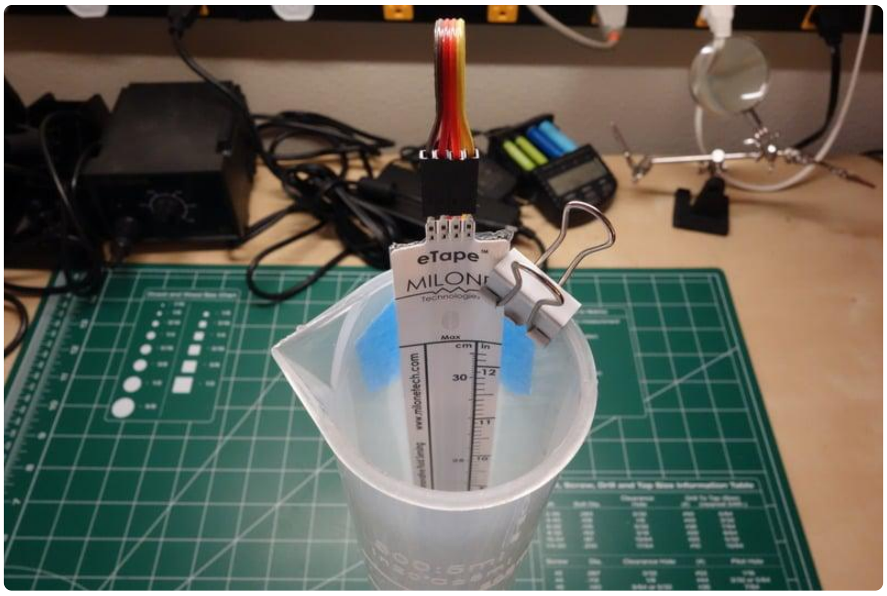




To attach the liquid sensor to a container, place the sensor against a wall of the container. Make sure the sensor hangs vertically with no bends or crimps. Only the top portion of the sensor above the MAX line can be attached to the container. The rest of the sensor must be allowed to hang freely so liquid can touch both the front and back of the sensor. Check the sensor datasheet () if you need more information about using or mounting the sensor.

Below you can see how I attached the sensor to a graduated cylinder with a binder clip. This clip works well as a temporary means of affixing the sensor. For something more permanent consider taping or gluing the top of the sensor to the container (remember only glue or tape the sensor above the MAX line!). Use a food safe adhesive such as silicone sealant () if necessary.

You can also see I bent the wires in a tall loop and taped them onto the back of the cylinder as a strain relief.



Calibration

In this step the Arduino sketch will be calibrated to report the volume of measured liquid. To get started, download the software for this project from the following link:

Download Software

Unzip the archive and load the LiquidSensor sketch in Arduino. This sketch will read the raw sensor resistance and help you calibrate for volume measurements.

At the top of the sketch you can adjust the following #define values based on your hardware:

- SERIES_RESISTOR - This is the value in ohms of the resistor you attached to the hardware in series with the sensor. This should be 560 unless you used a different resistor value.
- SENSOR_PIN - This is the analog input pin which is connected to the sensor.

Ignore the remaining #define values for calibration right now. Upload the sketch to your hardware and open the serial monitor at 115200 baud. You should see text like the following output every second:

```
Resistance: 1963.70 ohms
Calculated volume: 0.00000
```

The resistance value is the measured resistance of the liquid sensor. With no liquid touching the sensor it should read a value around 1500-2000 ohms (depending on the length of the sensor).

Try pressing the sensor in various locations to see how the resistance changes. If you touch near the top of the sensor at the MAX line you should see a resistance around 200-400 ohms.

To calibrate the sensor for measuring volume, you will need to measure the following:

- The resistance value with no liquid touching the sensor.
- The resistance value when a known volume of liquid is touching the sensor.
- The volume of liquid used to find the above resistance value.

For example I saw my sensor output a resistance of about 1963.7 ohms when no liquid was touching it. Then I filled the graduated cylinder with 500ml of water and saw the resistance was 512.81 ohms.

Once you've determined the above calibration values, update the #defines at the top of the sketch appropriately:

- ZERO_VOLUME_RESISTANCE - Set this to the resistance value with no liquid, in my case 1963.70.
- CALIBRATION_RESISTANCE - Set this to the resistance value with a known volume of liquid, in my case 512.81.
- CALIBRATION_VOLUME - Set this to the volume of liquid, in my case 500.00 milliliters (you can use other units, but milliliters are what the web application expects).

Save the sketch and upload again. Now you should see a calculated volume value output every second. Try filling the container with various levels of liquid to see how the calculated volume changes.

If you're not using an Arduino Yun, the software setup is done and you're now able to read the volume of liquid in the container. Consider adding a display, like this small OLED (http://adafru.it/931), to show the measured volume!

If you're using the Yun, continue on to learn about how to setup the Yun to display volume data on a web page.

Arduino Yun Setup

In this step you'll setup an Arduino Yun to display the measured volume on a web page that updates in real time.

Before you get started you will want to have your Yun connected to your wireless network, and be familiar with connecting to the Yun over SSH. Check out the following links for more information on these topics:

- Guide to the Arduino Yun ()
- SSH communication with the Arduino Yun ()

Sketch Setup

Load the YunSmartMeasuringCupSketch in Arduino and update the #define values at the top just like you did for the LiquidSensor sketch. Make sure to copy in the calibration values you determined from the previous step. Upload the sketch to the Yun.

Once the sketch is running it will wait for an internal connection on port 5678 of the Yun's Linux-based processor. When a client connects to this port, the Yun will start sending volume measurements continuously (about 5 times a second) over the connection. You can SSH to the Yun and execute the telnet localhost 5678 command to see this behavior for yourself (press Ctrl-C to quit telnet).

To send the volume data to a web page, a server will be run on the Yun which both serves a web page, and volume data read from port 5678.

Server Setup

In this step you will setup the Flask () web application framework as a server on the Arduino Yun. Flask is a simple python () framework for running web applications. The web page for this guide is written using Flask because of its support for streaming data through HTML5 server sent events (). Don't worry if you aren't familiar with Flask or web development, you can still follow the steps below to setup the server and web page.

First make sure the SD card is connected to the Yun, then connect to the Yun using SSH. Once connected execute the following commands to install pip, a python package manager:

```
opkg update
opkg install distribute
opkg install python-openssl
easy_install pip
```

Next execute the following command to create a directory on the SD card for storing python packages:

```
mkdir /mnt/sda1/python-packages
```

This folder will be used to install Flask and other python dependencies. By storing these dependencies on the SD card, there's less concern about using up the very limited storage space on the Yun (only 16 megabytes for the entire Linux installation!).

Finally execute the following command to install Flask:

```
pip install --target /mnt/sda1/python-packages flask
```

Now copy the YunSmartMeasuringCupServer folder from the software download onto the SD card of the Yun. You can use a file manager like Cyberduck () to transfer files to the Yun, or if on Mac or Linux execute a command like:

```
scp -r YunSmartMeasuringCupServer root@arduino.local:/mnt/sda1
```

Note that the above command should be executed on your computer and from the directory where the software was unzipped. The command also assumes the Yun is using the default name 'arduino'.

Usage

To start the server, in an SSH session with the Yun execute the following:

```
cd /mnt/sda1/YunSmartMeasuringCupServer
python YunSmartMeasuringCup.py
```

You should see a message such as the following which indicates Flask is now listening on port 5000:

```
* Running on http://0.0.0.0:5000/ ()
* Restarting with reloader
```

If you see an error or other failure, carefully check that all the above steps were followed.

With the server running, open a web browser to http://arduino.local:5000/ () (or change the arduino.local to the appropriate name of your Yun). If you're running from a device without mDNS support (like Android or Windows), replace arduino.local with the IP address of the Yun.

Once the web page loads you should see it connect to the server and start displaying volume measurements. Try adding liquid to the container to see the measurement update in real time! Note that only one person can be connected to the server at a time.

When you're done running the server, on the Yun SSH connection that started the server press Ctrl-C to stop Flask.

If you want the server to run automatically on boot, edit the /etc/rc.local file on the Yun and add this line:

```
python /mnt/sda1/YunSmartMeasuringCupServer/YunSmartMeasuringCup.py
```

Enjoy using your smart measuring cup!

Future Work

This project is an example of how to use an eTape liquid level sensor to build a smart web-enabled measuring cup. You can use the code from this guide to help get any sensor data onto a web page using an Arduino Yun.

Some interesting ways you might extend this project include:

- Log volume or liquid level data to a cloud service such as Google Docs () or Amazon DynamoDB ().
- Put a water level sensor in your home's sump pump and send an SMS message () when water is rising dangerously high!
- Make a web application to help you mix drinks () and other concoctions with the liquid sensor.

What can you think of to do with a liquid sensor and Arduino Yun?