Title: *IoT Water Leakage Indicator*

I. Summary

This project will keep you aware about leakage of water from pipe or tank. Leaks from pipes, plumbing fixtures and fittings are a significant source of water waste for many households. Research has shown that the typical home can lose 2,000 to 20,000 gallons of water per year due to leaks. Some leaks are obvious, such as dripping faucets and leaking water heaters. Unfortunately, many leaks go undetected for years because the source of the leak is not visible.

II. Objectives

1. To show the student how to use an Arduino (Micro-Controller)
2. To show the student how to use IoT (Internet of Things)

III. Industry-Based Applications

In the Nuclear Industry, many pipes and valves are used to clean and heat radioactive water. Eventually, these pumps, pipes, and valves begin to degrade during the Nuclear Plants lifetime. For example, the water is pressurized for cooling and heating which, in turn, allows for the water to be recycled.[1] Preventative measures are done daily, weekly, monthly, and annually to ensure that none of the water or contaminates are released into the water and to the public.

IV. Project Methodology

In this project, we will use an Arduino, which is a Micro-controller. Next, a transistor LM7805, ESP266 wife module, a 3.3-9 Volt power supply, Arduino software, and access to a *Smart server*. The idea of this project is to introduce students to alarm systems such

that they focus on preventative measure related to the scope of their project ideas or careers. This will teach students how to use and navigate a server, Micro-controller and transistors.

V. Project Procedure

Step 1. A 9V battery is used as a power supply. Also, the project requires a 7805-voltage regulator to receive 9 V potential which then feeds into the Arduino.

Figure 1-a.

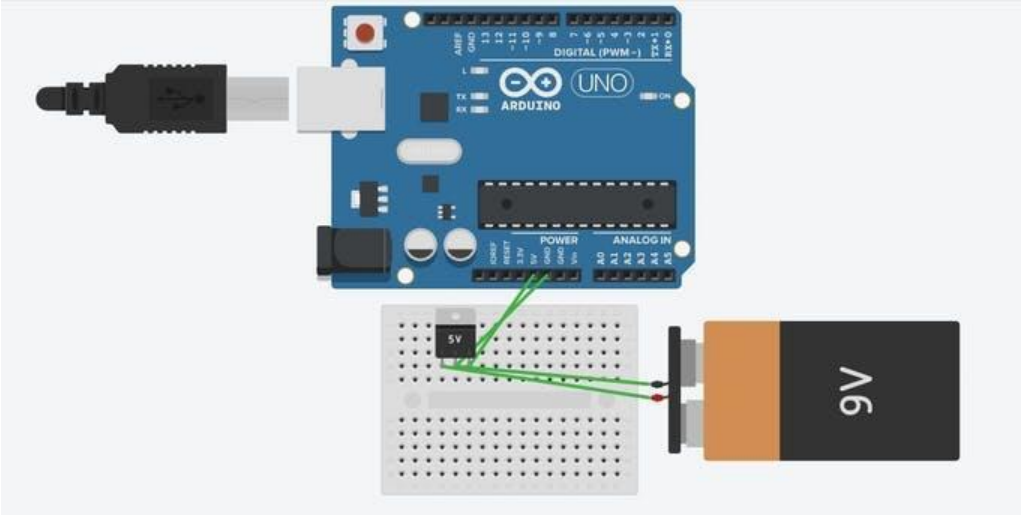
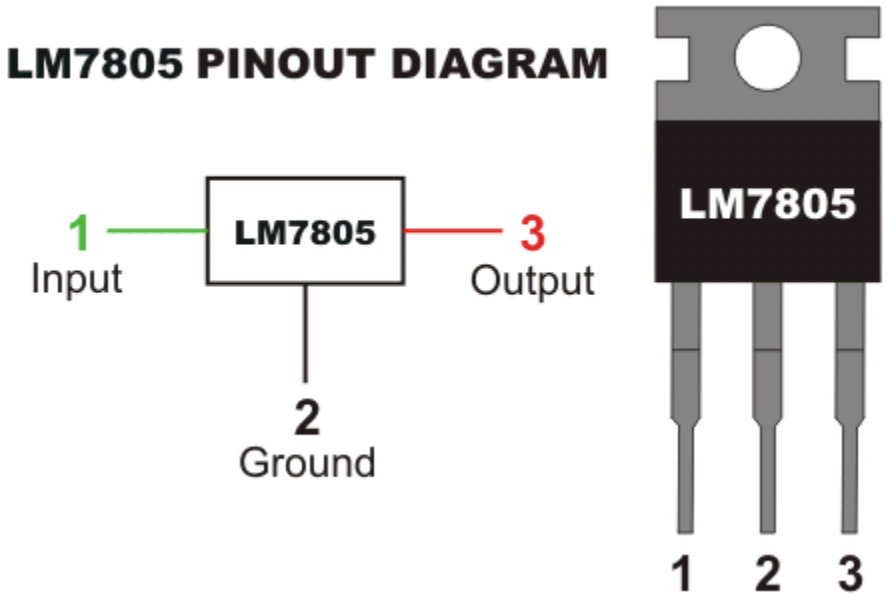


Figure 1-b.



Step 2. Next, the ESP266 is connected as a way of powering the device. The output of the Arduino in connected to the 3.3V to the ESP8266. The reason being, is that the ESP8266 only works with 3.3V and **NOT** the 5V.

Figure 2-a.

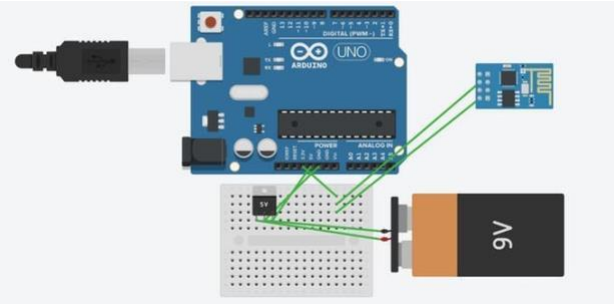
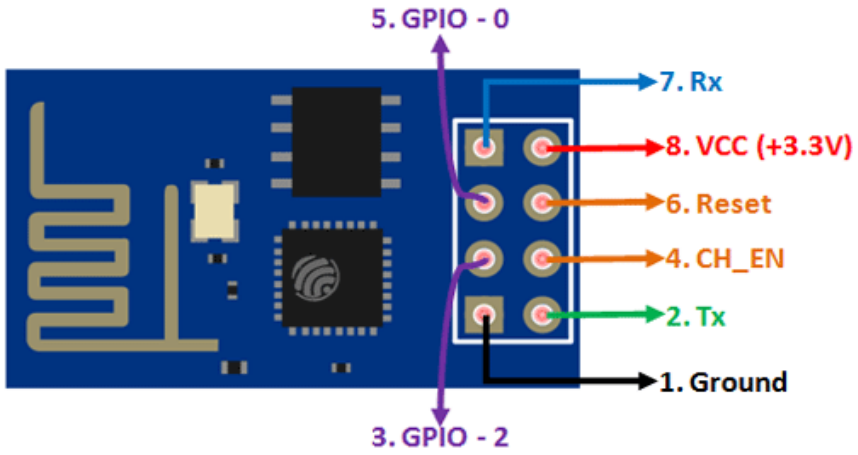
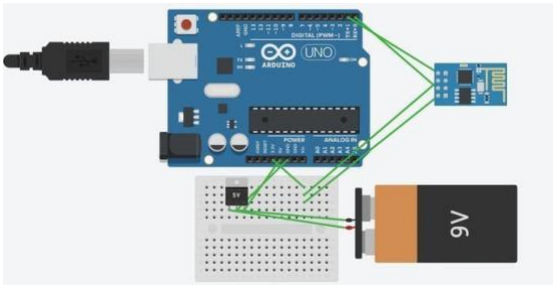


Figure 2-b. Esp8266 WIFI Module



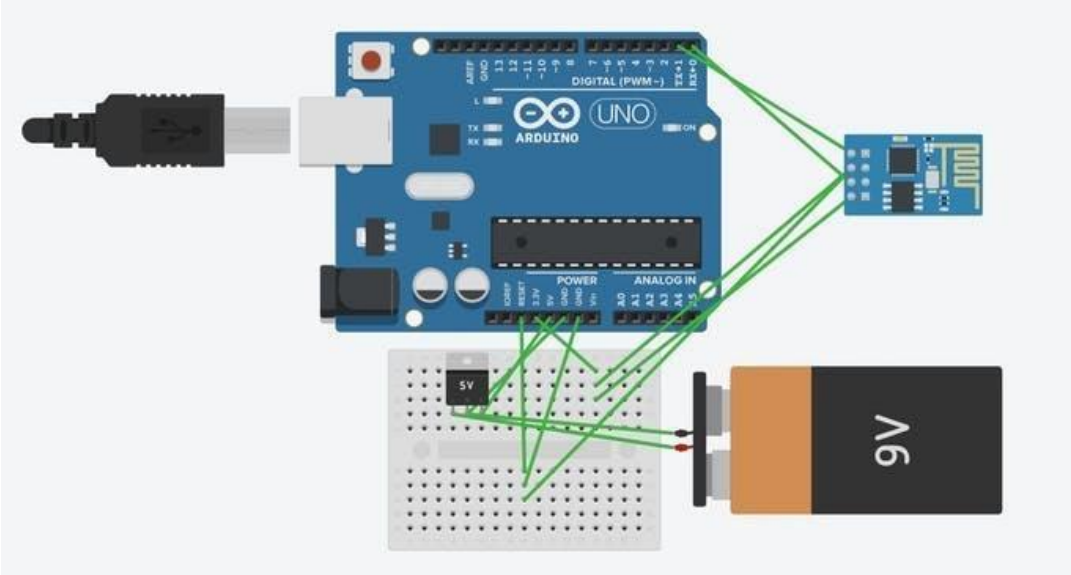
Step 3: This step is very important such that the student needs to make sure the Arduino is connected appropriately. This step is to connect the RX and TX. Connect the RXD pin of the Arduino to the RX pin of the ESP8266.Connect the TXD pin of the Arduino to the TX pin of the ESP. When we want two things to talk to each other over serial, we connect the TX pin of one to the RX of the other (send goes to receive and the opposite). Here we do not have the Arduino talk to the ESP8266 though, our computer is talking to it via the Arduino.

Figure 3-a.



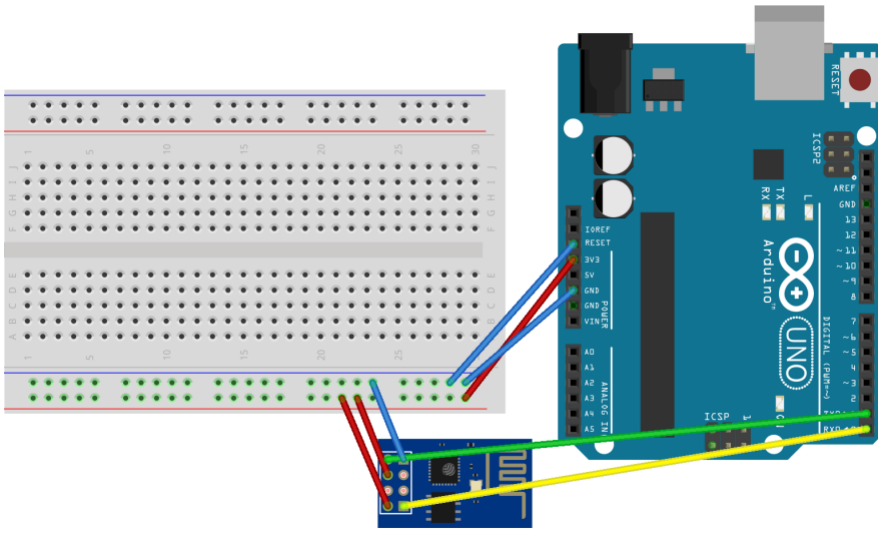
Step 4: This step introduces the GND and RST. Connect the RES or RESET pin, when you ground the reset pin, the Arduino works as a dumb USB to serial connector, which is what we want to talk to the ESP8266.

Figure 4-a.



Step 5a: This step includes two parts. Next we connect Arduino and water sensor. We connect Arduino VCC and GND to water sensor VCC and GND. Arduino digital Pin 2 is connected to digital out pin of water sensor. The following link will show you about the sensor:

https://youtu.be/7AzMLL5yXak



5b: Upload the code to the downloaded software.

```
#include <Esp8266EasyIoT.h>
#include <SoftwareSerial.h>

Esp8266EasyIoT esp;

SoftwareSerial serialEsp(10, 11);

#define LEAK_PIN  2  // Arduino Digital I/O pin number
#define CHILD_ID_LEAK 0

Esp8266EasyIoTMsg msgLeak(CHILD_ID_LEAK, V_DIGITAL_VALUE);
//Esp8266EasyIoTMsg msgHum(CHILD_ID_LEAK, V_LEAK); // supported in esp >= V1.1 lib

int lastLeakValue = -1;

void setup()
{
  serialEsp.begin(9600);
  Serial.begin(115200);

  Serial.println("EasyIoTEsp init");
  esp.begin(NULL, 3, &serialEsp, &Serial);

  pinMode(LEAK_PIN, INPUT);

  esp.present(CHILD_ID_LEAK, S_LEAK);
}

void loop()
{
  esp.process();

  // Read digital pin value
  int leakValue = digitalRead(LEAK_PIN);
  // send if changed
```

```
  if (leakValue != lastLeakValue) {
    Serial.println(leakValue);
    esp.send(msgLeak.set(leakValue==0?0:1));
    lastLeakValue = leakValue;
  }
}
```

Code copied from Arduino:

```
void setup() {
  // put your setup code here, to run once:


#include <Esp8266EasyIoT.h>
#include <SoftwareSerial.h>


Esp8266EasyIoT esp;


SoftwareSerial serialEsp(10, 11);


#define LEAK_PIN  2  // Arduino Digital I/O pin number
#define CHILD_ID_LEAK 0


Esp8266EasyIoTMsg msgLeak(CHILD_ID_LEAK, V_DIGITAL_VALUE);
//Esp8266EasyIoTMsg msgHum(CHILD_ID_LEAK, V_LEAK); // supported in esp >= V1.1 lib


int lastLeakValue = -1;


void setup()
{
  serialEsp.begin(9600);
  Serial.begin(115200);


  Serial.println("EasyIoTEsp init");
  esp.begin(NULL, 3, &serialEsp, &Serial);
```

```
  pinMode(LEAK_PIN, INPUT);

  esp.present(CHILD_ID_LEAK, S_LEAK);
}

void loop()
{
  esp.process();

  // Read digital pin value
  int leakValue = digitalRead(LEAK_PIN);
  // send if changed
  if (leakValue != lastLeakValue) {
    Serial.println(leakValue);
    esp.send(msgLeak.set(leakValue==0?0:1));
    lastLeakValue = leakValue;
  }
}
```

Step 6: Connect to an online or on-campus server Such as *EasyIOT* server. [2]

This section will describe in detail how your project will be performed. You need to provide clear pictures and videos' links. You may start by a sub-title telling the components and explain the meaning of each symbol you used and what is/are the purpose/s of each component.

VI. References

1. https://www.nuclear-power.net/reactor-coolant-pump/
2. https://iot-playground.com/download

VII. Appendix

1. 7805 transistor Data Sheet

   http://ee-classes.usc.edu/ee459/library/datasheets/LM7805.pdf

2. Arduino UNO Data Sheet

   https://www.farnell.com/datasheets/1682209.pdf

3. Genuino Data Sheet

   https://www.mouser.com/catalog/specsheets/arduinocc_GenuinoProducts Brief.pdf

4. ESP8266 Data Sheet

   https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

   https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf

---

Please Note that if you are using a code, it should be provided in the original format with this report.