

Plastic Bottle Rewarder using IOT

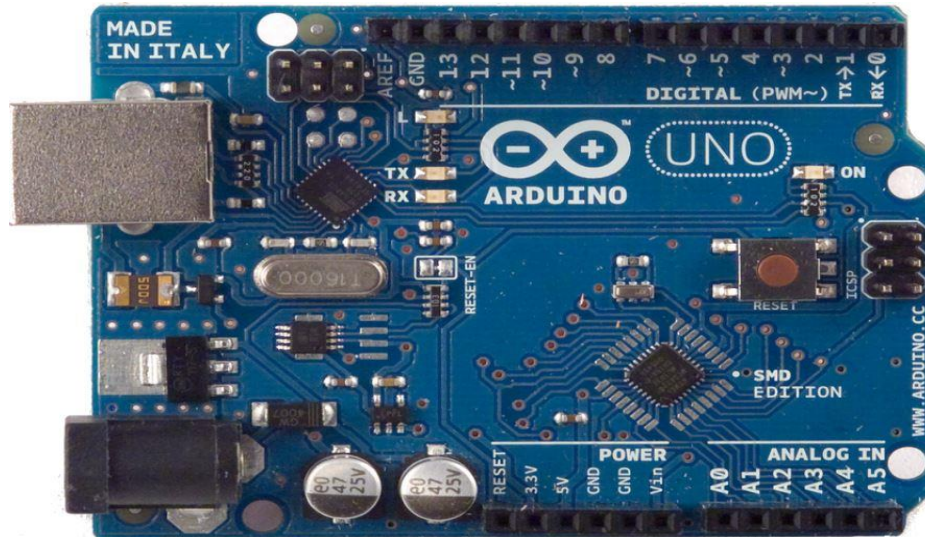
This project is aimed and designed to recycle the plastic bottles which are using in a vast way. The designed project will be implemented in the busiest areas like railway stations, bus stops, airports etc. where the usage of bottles are more. The user scans the QR code and fill the email and phone number. After submitting, he/she need to throw a plastic bottle into the crusher. If it is a plastic bottle, he/she will get an email which rewards with some points. Thereafter, he/she can claim the rewarded points to get the money.



2.Things used in this project

2.1.Hardware Components and their description:

2.1.1 Arduino:

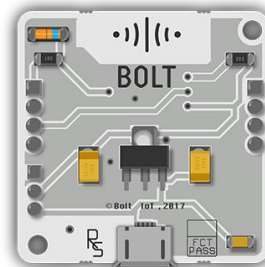
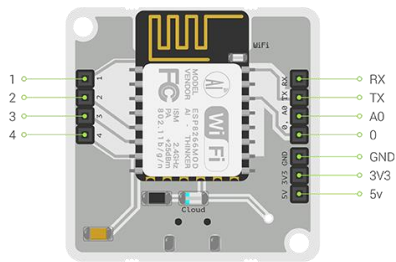


An Arduino is an open-source microcontroller development board based on easy-to-use hardware and software. Simply, you can use the Arduino to read sensors and control things like motors and lights. You can tell your board what to do by sending a set of instructions to the microcontroller on the board which can then interact with things in the real world. With this, you can make devices which respond and react to the world at large.

Features:

1. Microcontroller: Atmega328
2. Operating Voltage: 5
3. Input Voltage (recommended): 7-12V
4. Input Voltage (limits): 6-20V
5. Digital I/O Pins: 14 (of which 6 provide PWM output)
6. Analog Input Pins: 6
7. DC Current per I/O Pin: 40 mA
8. DC Current for 3.3V Pin: 50 mA
9. Flash Memory: 32 KB (Atmega328) of which 0.5 KB used by bootloader
10. SRAM: 2 KB (Atmega328)
11. EEPROM: 1 KB (Atmega328)

2.1.2.Bolt(ESP8266):



Parameters

Details

Connectivity and Processing Module

ESP8266 with custom firmware

MCU

32-bit RISC CPU: Tensilica Xtensa LX106

Power

5V/1A DC via Micro-USB port or 5V and GND pins

Operating Voltage

3.3V

CPU Clock Frequency

80 MHz

MCU Internal Memory

64 KB of instruction RAM; 96KB of data RAM

MCU External Memory

4 MB Flash memory [QSPI]

GPIO pins

5 Digital pins [3.3V logic]

ADC

1 pin 10 bit ADC [0-1V input]

PWM

All 5 Digital pins capable of PWM [Software PWM]

Connectivity:

| Parameters | Details |
|------------|---|
| WiFi | 802.11 b/g/n Automatic AP mode if not connected to WiFi WEP/WPA/WPA2 authentication Only works with 2.4 GHz WiFi |

| | |
|-------|--|
| UART | 8-N-1 3.3V TTL UART [using TX, RX, GND pins] [2400,4800, 9600,19200 baudrate] |
| Cloud | Default: Bolt Cloud Optional: Custom cloud using Bolt APIs |

LED Indicators

| Parameter | Details |
|-------------------------------------|---|
| WiFi LED - WiFi connectivity | 1) Slow blinking: Trying to find and connect to WiFi network 2) Fast blinking: User has connected via Bolt IoT app for setup 3) Stable: Connected to WiFi |
| Cloud LED – Bolt Cloud connectivity | 1) Stable: Connected to Bolt Cloud 2) Off: Not connected to Bolt Cloud 3) Dim: Insufficient power/ incorrect boot |

2.1.3. IR Sensor:



An infrared sensor is an electronic instrument that is used to sense certain characteristics of its surroundings. It does this by either emitting or detecting infrared radiation. The infrared waves typically have wavelengths between 0.75 and 1000µm.

2.1.4. Servo Motor:



A servo motor is an electrical device which can push or rotate an object with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a servo mechanism, can rotate up to 180° .

2.1.5. LDR



A photoresistor (or light-dependent resistor) is a light-controlled variable resistor. The resistance of a photoresistor decreases with increasing incident light intensity. A photoresistor can have a resistance as high as several megaohms (M Ω), while in the light, a photoresistor can have a resistance as low as a few hundred ohms.

2.1.6. Buzzer



A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke.

2.2.Softwares and programming languages used:

2.2.1. Arduino IDE:

Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.

2.2.2. Python3:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

2.2.3. Mailgun:

Mailgun is the all-in-one, intelligent email delivery platform trusted by businesses, loved by developers. Mailgun is how modern companies work with email. The platform's ease of use, world-class support, and powerful APIs empower smart development teams to reach real customers at scale with a data-driven approach so their organization can grow faster.

2.2.4. Google sheets and forms:

Google Sheets is a spreadsheet program included as part of a free, web-based software office suite offered by Google within its Google Drive service. The service also includes Google Docs and Google Slides, a word processor and presentation program respectively. Google Forms is a survey administration app that is included in the Google Drive office suite along with Google Docs, Google Sheets, and Google Slides. Forms features all of the collaboration and sharing features found in Docs, Sheets, and Slides

2.2.5. QR code generator:

A QR Code is a two-dimensional barcode that is readable by smartphones. It allows to encode over 4000 characters in a two dimensional barcode. QR Codes may be used to display text to the user, to open a URL, save a contact to the address book or to compose text messages. To read QR Codes with your smartphone, you need an appropriate software installed on your phone. For Android-based devices, you can use [Barcode Scanner](#) by ZXing. On iOS-Devices like iPhones there are also QR Code readers available on the AppStore, for Example i-nigma. On Firefox OS try QR code scanner. If you use a modern web browser (recent versions of Firefox or Chrome) and have a webcam, this site allows you to scan QR

Codes with your web browser. You can use this to send a link from a phone to your desktop browser .

2.2.6. APIs used:

An *application program interface* (API) is a set of routines, protocols and tools for building software applications. Basically, an API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface components. A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together. In simple, it is a messenger between two applications.

a) Mailgun API:

The Mailgun API is built on HTTP. Mailgun has published Libraries for various languages. You may use mailgun libraries, or your favorite HTTP/REST library available for programming language, to make HTTP calls to Mailgun.

b) Bolt API:

The Bolt Cloud API provides an interface for communication between the Bolt devices and any 3rd party system e.g. mobile application, web server, python programs etc. The API contains very intuitive control, monitoring, communication and utility functions for the Bolt Devices connected to your account. The Bolt Cloud API uses HTTP protocol for the communication and uses the HTTP GET and HTTP POST methods. Hence users can execute actions and retrieve information from Bolt devices programmatically using conventional HTTP requests.

Here are a few use cases of the API:

- Use the API in native apps on iOS and Android to control and monitor Bolt devices over the Internet.
- Pull collect sensor data connect to Bolt device, to any other cloud to run your custom AI algorithms and analytics.
- Connect Bolt Cloud to any VPS (Virtual Private Server) and run your code in any language of your choice. Refer sample codes.
- Remote Operating System: Using the API, Bolt devices can work like a board with an OS i.e. similar to Raspberry Pi or Beagle Bone, with the exception of the OS, which in this case, will reside on a remote VPS (Virtual Private Server). The Bolt will receive data from the sensors and push to the VPS with a Linux OS. The processing will take place on the VPS and it will push the commands to control motors, LEDs, and actuators to the Bolt device. You can use all the features of a Linux OS in this kind of a system.

c) Google Sheet API:

Google Sheets is an online spreadsheet app that lets users create and format spreadsheets and simultaneously work with other people. Google Sheets isn't only for consumers:

it's used every day by businesses and schools to manage spreadsheet data. With the new Sheets API v4 and Sheets add-ons, that data can be accessed by code as well as users.

d) Google Drive API:

Create apps that read, write, and sync files in Google Drive. The Drive API allows us to create mobile, web, and desktop apps that read, write, and sync files in our Google Drive.

2.2.7. Libraries used:

a) gspread Library

gspread is a Python API for Google Sheets.

Features:

- Google Sheets API v4.
- Open a spreadsheet by its **title** or **url**.
- Extract range, entire row or column values.
- Python 3 support.

b) oauth2client Library

OAuth is a standard that applications (*and the developers who love them*) can use to provide client applications with “secure delegated access”. OAuth works over HTTP and authorizes Devices, APIs, Servers and Applications with access tokens rather than credentials, which we will go over in depth below.

There are two versions of OAuth: OAuth 1.0a and OAuth2. These specifications are *completely different* from one another, and cannot be used together: there is *no* backwards compatibility between them.

OAuth is basically a protocol that supports authorization workflows. What this means is that it gives you a way to ensure that a specific user has permissions to do something.

c) Serial Library

The Serial library reads and writes data to and from external devices one byte at a time. It allows two computers to send and receive data. This library has the flexibility to communicate with custom microcontroller devices and to use them as the input or output to Processing programs. The serial port is a nine pin I/O port that exists on many PCs and can be emulated through USB.

This module encapsulates the access for the serial port. It provides backends for Python running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython. The module named “serial” automatically selects the appropriate backend.

Features

Same class based interface on all supported platforms. Access to the port settings through Python properties. File like API with “read” and “write” (“readline” etc. also supported). The files in this package are 100% pure Python.

d) time Library

The Python time module provides many ways of representing time in code, such as objects, numbers, and strings. It also provides functionality other than representing time, like waiting during code execution and measuring the efficiency of your code.

Although this module is always available, not all functions are available on all platforms. Most of the functions defined in this module call platform C library functions with the same name. It may sometimes be helpful to consult the platform documentation, because the semantics of these functions varies among platforms.

e) boltiot Library

boltiot is a python library which is used to interact with the bolt module using the APIs which are unique for every device. The Bolt APIs are used to connect the Bolt based hardware and the Bolt Cloud to any third party system.

f) json Library

JavaScript Object Notation is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types. It is a very common data format, with a diverse range of applications, such as serving as replacement for XML in AJAX systems.

The json library can parse JSON from strings or files. The library parses JSON into a Python dictionary or list. It can also convert Python dictionaries or lists into JSON strings.

JSON is a language-independent data format. It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON-format data. The official Internet media type application/json. JSON filenames use the extension .json.

3.Story

3.1. Introduction

A million plastic bottles are bought around the world every minute. Nearly, 20,000 bottles are sold every second.

More than 480 billion plastic drinking bottles were sold in 2016 across the world. But, fewer than half of the bottles were collected for recycling and just 7% of those collected were turned into new bottles. Instead most plastic bottles produced end up in landfill or in the ocean.

To avoid the bottles to end up in the landfills , oceans and to increase the recycling, this project collects the crushed plastic from the busiest areas for recycling.

3.2. Steps for building the project

Step 1: The user has to scan the QR CODE available on the machine. QR code is shown below.



Step 2: After scanning the QR code, in the google form the user has to enter his email and phone number. This data will be stored in the google sheets and it will be accessed in our python code. For that we need a google configuration file which is confidential.

Code for accessing data from google sheets

```
import gspread
from oauth2client.service_account import ServiceAccountCredentials

# how we interact with our data in Google Sheets
scope=["https://spreadsheets.google.com/feeds",\
"https://www.googleapis.com/auth/spreadsheets",\
"https://www.googleapis.com/auth/drive.file",\
"https://www.googleapis.com/auth/drive"]

#creating client to interact with Google Sheets
credentials=ServiceAccountCredentials.\
from_json_keyfile_name("credentials.json",scope)
```

```
client=gspread.authorize(credentials)
sheet=client.open("project1").sheet1
```

```
#j is a variable where the email is stored in google sheets
cello2=sheet.cell(j,2).value
```

Step 3:After detecting email,python code sends a byte of data to arduino to activate the circuit.

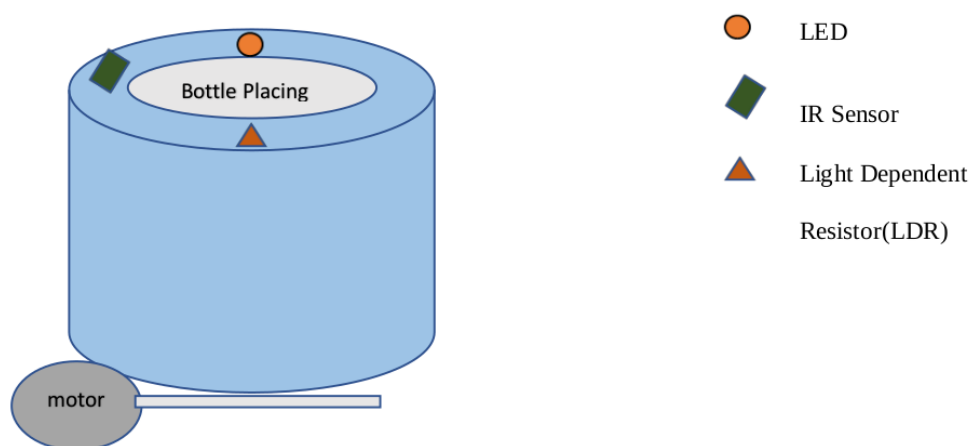
Code for sending data through USB port

```
ArduinoSerial=serial.Serial('/dev/ttyACM0',9600)
time.sleep(3)
ArduinoSerial.write(b'1')
```

Step 4:After submitting email,the user has to throw the plastic bottle into the machine.Arduino checks whether it is a bottle or not using a circuit and sends the data through USB port which will be accessed in the python code.

About Bottle Detection:

Firstly, IR sensor will detect the object nearer to it.If there is an object, LED will emit light.Generally,Plastic bottles are transparent.So,if light passes through it.There 1 will be a LDR to the opposite side of LED.LDR will sense the intensity of light.As,bottle is transparent the intensity of light is more at the LDR when compared to the opaque objects like stones.If it is bottle then,the Servo motor will open the barrier to make a way for bottle into the Crusher.



Arduino code for detecting the bottle and sending data through USB:

```
#include <Servo.h>
//defining variables
#define ir 4
#define buzzer 8
#define led 7
#define ldr A0
int i_r=0;
int angle=120;
Servo servom;
void setup() {
    //describing pinmodes of all the pins
    Serial.begin(9600);
    pinMode(ir,INPUT);
    pinMode(led,OUTPUT);
    pinMode(ldr,INPUT);
    pinMode(buzzer,OUTPUT);
    servom.attach(11);
    servom.write(angle);
}

void condition(int i_r,int l_d_r){
    int sth=0;
    if(i_r==0 && l_d_r >=800){    //6.condition for opening motor

//7.instructions for motor functioning based on our application.
        for(angle=120;angle<=0;angle-=1){
            servom.write(angle);
            delay(10);}
        for(angle=0;angle<=120;angle+=1){
            servom.write(angle);
            delay(10);}
        sth=11;
        delay(500);
        Serial.println(sth);}
    else if(i_r==0 && l_d_r<700){    //8.condition for activating
                                    Buzzer.which means thrown one is not a
                                    bottle.

        Serial.println(2);
        digitalWrite(buzzer,HIGH);
        delay(1000);
        digitalWrite(buzzer,LOW);
    }
    else{
        //9.data sending through USB port which is accessed in python code.
        Serial.println(2);
    }
}
```

```

}
void func(){
    i_r=digitalRead(ir);
    if(i_r==0){
        digitalWrite(led,HIGH);
    }
    else{
        digitalWrite(led,LOW);
    }
    delay(1000);
    int l_d_r=analogRead(ldr);
    condition(i_r,l_d_r);
}

void loop(){
    while(Serial.available()>0){
        func();
        break;
    }
    //below code is for functioning the machine without taking any data from the
    user as he/she is not interested for money.
    if(Serial.available()==0){
        i_r=digitalRead(ir);
        if(i_r==0){
            digitalWrite(led,HIGH);
        }
        else{
            digitalWrite(led,LOW);
        }
        delay(1000);
        int l_d_r=analogRead(ldr);
        if(i_r==0 && l_d_r >=800){
            for(angle=120;angle<=0;angle-=1){
                servom.write(angle);
                delay(10);}
            for(angle=0;angle<=120;angle+=1){
                servom.write(angle);
                delay(10);}
        }
        else if(i_r==0 && l_d_r<700){
            digitalWrite(buzzer,HIGH);
            delay(1000);
            digitalWrite(buzzer,LOW);
        }
        delay(2000);
    }
}

```

Step 5: In python code, data will be taken through USB. If it is a plastic bottle, an email is sent to the user with some reward points. For that, Bolt ESP8266 module is used. It has a configuration file for accessing it, which is confidential.

Python code for taking data from arduino through USB:

```
ard_data.append(ArduinoSerial.readline())
```

data from arduino will be either 11 or 2. If thrown one is bottle, it will be 11 or else, it will be 2. So, if data is 11, email will be sent.

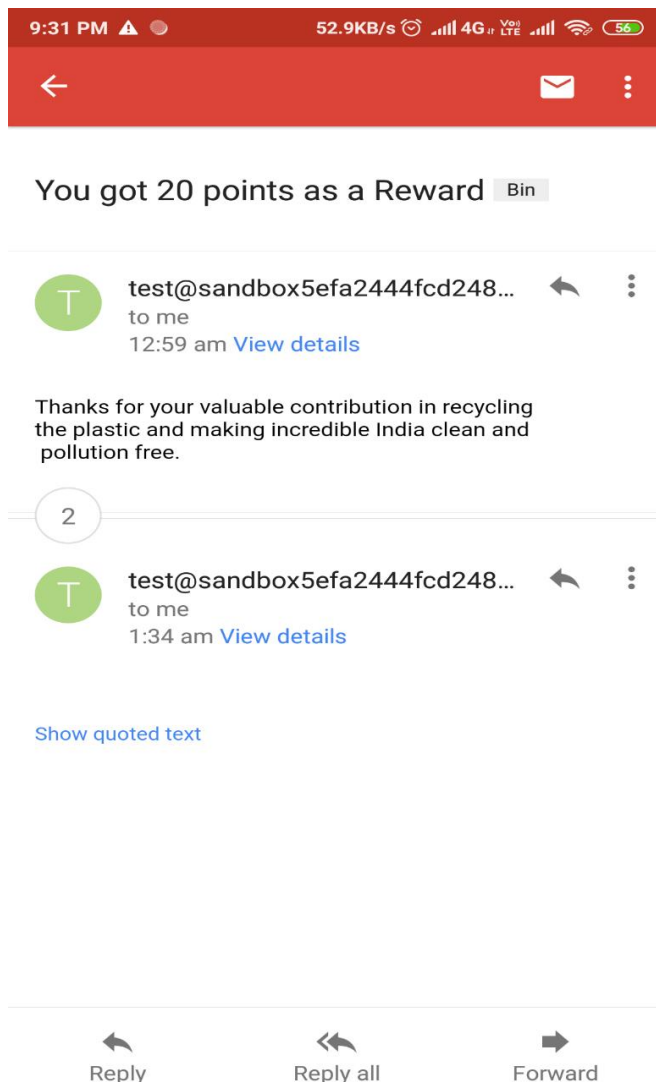
Python code for sending mail through Bolt:

```
import email_conf
from boltiot import Email, Bolt
import json, time

mybolt = Bolt(email_conf.API_KEY, email_conf.DEVICE_ID)

def mail_func(email_id):
    mailer = Email(email_conf.MAILGUN_API_KEY, \
        email_conf.SANDBOX_URL, email_conf.SENDER_EMAIL, email_id)
    try:
        print("Making request to Mailgun to send an email")
        response = mailer.send_email("You got 20 points as a Reward", \
            "Thank you for your valuable contribution in recycling \
the plastic and making incredible india clean and pollution free.")
        response_text = json.loads(response.text)
        print("Response received from Mailgun is: " + \
            str(response_text['message']))
    except Exception as e:
        print ("Error occured: Below are the details")
        print (e)
```

Step 6: The user can claim these reward points to get amount into his account. Below image is the screenshot of message received to the gmail.



Note: If the user is in hurry or not interested for money but, wants to be a part of recycling, he/she can just throw the plastic bottle into the crusher where this condition is included in the arduino code itself.

Schematics:

