SOFE 4610U: Design and Analysis of IoT

Proposal: Smart Trash Can

Jane Coralde, 100660214

Ramin Sami, 100655257

Zainab Rehan, 100653955

Amin Khakpour , 100669547

10/19/2021

Project Description

Waste management is one of the vital functions of a smart city. Current mechanisms do not take into account the actual fill level from containers, and instead schedule collection on predefined intervals. Since the generation of trash is dynamic in nature, it often results in some trash cans getting overloaded and leaking while others remain almost empty.

An overflowing trash can be not only aesthetically unpleasant, it could also be a health hazard for communities, especially in warm environments. Our project, Smart Trash Can, is aimed towards the visually impaired in order to make their day to day lives a little easier. Smart Trash Can is a smart trash can designed to measure the capacity of a trash can as it's being filled up. Once the trash inside reaches a certain capacity within the can, users will either be notified through a mobile application that they must empty the Smart Trash Can. Alternatively, users can be alerted that Smart Trash Can has reached capacity with the use of LED lights on top of the can or through an audio cue such as a beep or alarm. Users can also check the capacity level through the mobile application

Last, but not least, the traffic congestion due to slow moving vehicles like trash pickup trucks is really annoying. Wouldn't it be nice if the vehicles knew in advance which trash can should be picked and which could be skipped? You could build one right away. Let's get started.

The proposed solution will use the following mechanisms:

- Ultrasonic sensor attached in lid of trash can to calculate trash(level).
- NodeMCU Wifi-Module is the base controller.
- Message alert system will be implemented using MQTT Protocol.
- Continuous data collection and sending of data in the cloud

Architecture

MQTT Communication system - Raspberry Pi to set up an MQTT Broker. Our team chose Mosquitto as MQTT software for the broker. This is used to receive MQTT data.

Using micro controllers and sensors to reduce the cost - also to add that we utilized the given hardware in class. Sensors proposed are ultrasonic and sensor nodeMCU.

NodeMCU:

- Enable Wi-Fi and establish MQTT connection.
  - In Arduino IDE, upload code with configuration of network, mqtt server information, authentication, and subscribing to MQTT topic
- Enable MQTT publish to topic to send data.

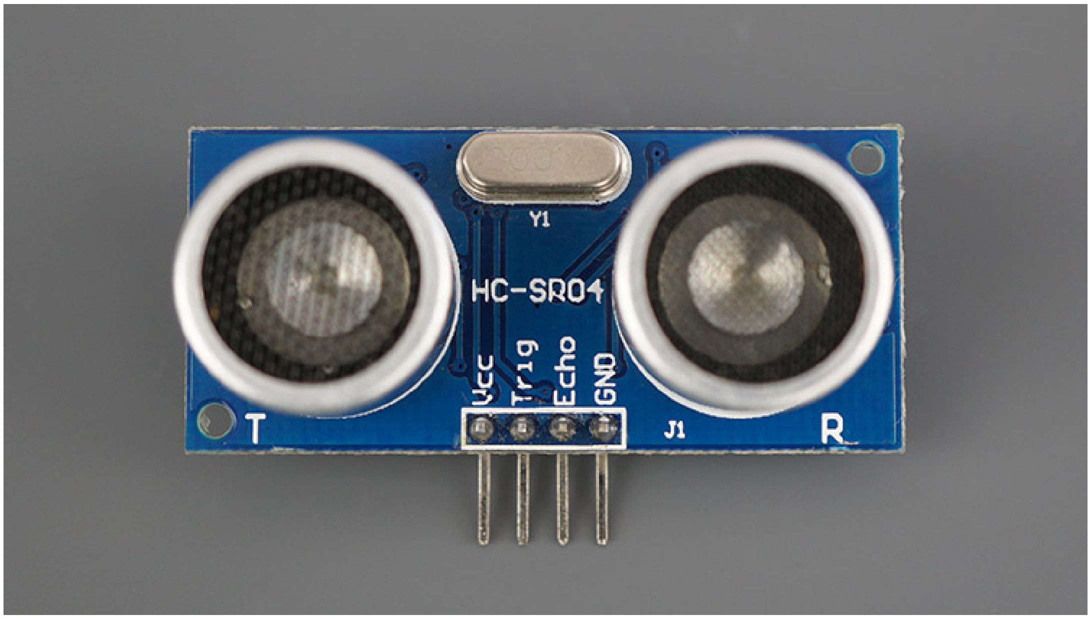
Ultrasonic sensor:

- Works on 5V (NodeMCU supports 3.3V)

- Pins utilized in code to get data stored as strings. Pinout is as follows, please see figure below

VCC	5V
Trig	Trigger Input Pin
Echo	Echo Output Pin
GND	Common GND

- Strings in the form of message (or payload) will be sent via MQTT to the broker.
    - Added in code is the mechanism to send alerts to users based on level



Functional and Non-functional Requirements

Functional Requirements

R1	The device shall record the depth of a trash can at a 60 minute interval.
R2	The device shall notify the user through the app once the trash can reaches capacity.
R3	The device shall enable a user to view how full a trash can is on the app.
R4	The device shall contain an LED which turns red once capacity is reached
R5	The device shall contain an LED which turns green if capacity is not reached
R6	The device shall produce beeps in predetermined intervals as an audio cue, if capacity is reached

Non-functional Requirements

Availability

R7	The application shall have high availability
R8	The application shall not have unexpected downtime
R9	The application shall have downtime only during low-intensity hours

User Friendliness

R10	The application will be user friendly, particularly toward visually-impaired users
R11	The application shall have a consistent User Interface
R12	The application shall have a User Interface which is intuitive to its users, particularly toward its visually-impaired users

Accessibility

R13	The application shall be highly accessible
R14	The application shall be accessible to visually-impaired users, who should easily be able to navigate the system and have access to all content and functionality