

Project Code:

```
#include <WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

#include <ArduinoJson.h>

const int trigPin = 5;

const int echoPin = 18;


//define sound speed in cm/uS
#define SOUND_SPEED 0.034

#define CM_TO_INCH 0.393701

#define WIFI_SSID "homespot"

#define WIFI_PASSWORD "betheforce"

// Telegram BOT Token (Get from Botfather)
#define BOT_TOKEN "5362043814:AAHWbkyjg3RW06Ztcmbb1x3ptCUydo3l8iw"


// Use @myidbot (IDBot) to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
// message you

#define CHAT_ID "1794700596"

WiFiClientSecure secured_client;

UniversalTelegramBot bot(BOT_TOKEN, secured_client);


long duration;

float distanceCm;

float distanceInch;


void setup() {

  Serial.begin(115200); // Starts the serial communication

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
```

```

Serial.println();

// attempt to connect to Wifi network:

Serial.print("Connecting to Wifi SSID ");
Serial.print(WIFI_SSID);
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
secured_client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for
api.telegram.org
while (WiFi.status() != WL_CONNECTED)
{
    Serial.print(".");
    delay(500);
}
Serial.print("\nWiFi connected. IP address: ");
Serial.println(WiFi.localIP());

Serial.print("Retrieving time: ");
configTime(0, 0, "pool.ntp.org"); // get UTC time via NTP
time_t now = time(nullptr);
while (now < 24 * 3600)
{
    Serial.print(".");
    delay(100);
    now = time(nullptr);
}
Serial.println(now);
bot.sendMessage(CHAT_ID, "Smart Dustbin Connected to Cloud", "");
}

void loop() {
    // Clears the trigPin

```

```

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);


// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);


// Calculate the distance
distanceCm = duration * SOUND_SPEED/2;


// Convert to inches
distanceInch = distanceCm * CM_TO_INCH;


// Prints the distance in the Serial Monitor
Serial.print("Distance (cm): ");
Serial.println(distanceCm);
Serial.print("Distance (inch): ");
Serial.println(distanceInch);

// bot.sendMessage(CHAT_ID, "Bot started up", "");
// if(distanceCm < 17) bot.sendMessage(CHAT_ID, "Dustbin is empty", "");
if(distanceCm < 5){
  bot.sendMessage(CHAT_ID, "Dustbin NO : 1 is Full", "");}

delay(5000);
}

```