University Sidi Mohamed Ben Abdellah
National School of Applied Sciences of Fez
Embedded Systems and Industrial Computing Engineering

Precaution against corona: Smart soap pump bottle

Mohammed DRIOUECHE

April 13, 2020

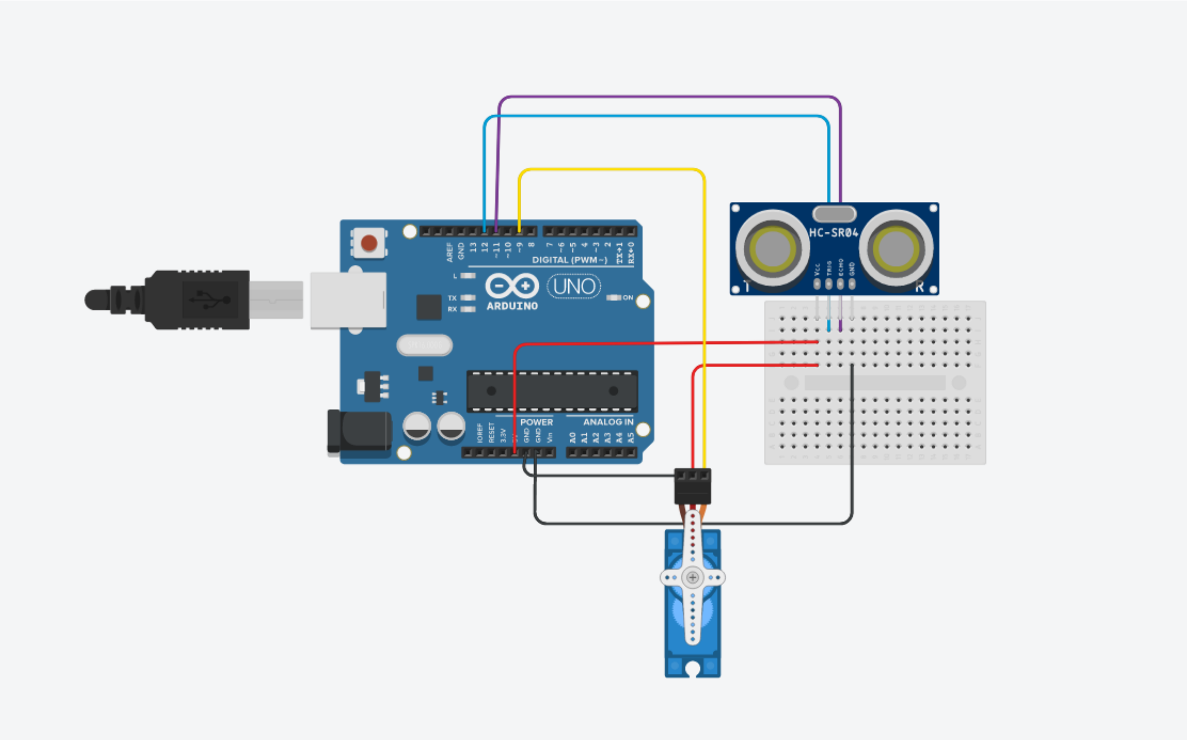
Components and supplies

- 1x Arduino UNO
- 1x Servo Motor
- Ultrasonic Sensor (hc-sr04)

Apps and online services

- Arduino IDE

Schematics



Code

Importing libraries

```
#include <Arduino_FreeRTOS.h>
#include <Servo.h>
```

Defining Arduino pins numbers

```
const int trigPin = 12;
const int echoPin = 11;
const int servoPin = 9;
```

Defining global variables

```
long duration;
int distance; //Formula: Distance = (Speed of sound   duration)/2
Servo myservo;  //Servo object
TaskHandle_t xHandle ;
```

Defining tasks prototype

```
void TaskUltrasonicSensor( void *pvParameters );
void TaskServoMotor( void *pvParameters );
```

Setup

```
void setup(){

  //Serial.begin(9600);

  //Servo motor is attached to pin 9
  myservo.attach(servoPin);
  myservo.write(0); //initialize servo motor

  // Now set up two Tasks to run independently.
  xTaskCreate(
    TaskUltrasonicSensor
    ,  ”Ultrasonic␣Sensor”  // A name just for humans
    ,  128
    // This stack size can be checked & adjusted by reading the Stack Highwater
    ,  NULL
    ,  2
    // Priority, with 3 (configMAX_PRIORITIES - 1) being the highest,
    //and 0 being the lowest.
    ,  &xHandle );

  xTaskCreate(
    TaskServoMotor
    ,  ”Servo␣Motor”  // A name just for humans
    ,  128
    // This stack size can be checked & adjusted by reading the Stack Highwater
    ,  NULL
    ,  1
    // Priority, with 3 (configMAX_PRIORITIES - 1) being the highest,
    //and 0 being the lowest.
    ,  NULL );

}
```

Loop

```
void loop(){}
```

TaskUltrasonicSensor

```
void TaskUltrasonicSensor(void *pvParameters){
  (void) pvParameters;

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

  for (;;) // A Task shall never return or exit.
  {
    // Clears the trigPin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    // 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);
    // Calculating the distance
    distance= duration*0.034/2;

    /*
    Serial.print(”Distance from the object = ”);
    Serial.print(distance);
    Serial.println(” cm”);
    */

    if(distance > 0 && distance<10){
      //suspend the task itself
      vTaskSuspend (NULL);
    }else {
      //Do nothing
      myservo.write(0);
    }
    vTaskDelay(1);  // one tick delay (15ms) in between reads for stability
  }
}
```

TaskUltrasonicSensor

```
void TaskServoMotor(void *pvParameters){
  (void) pvParameters;

  //Serial.println(” Execute...”);

  BaseType_t xYieldRequired;

  //push the pump
  myservo.write(90);

  //Resume the suspended task
  xYieldRequired = xTaskResumeFromISR(xHandle) ;
  if( xYieldRequired == pdTRUE ){
    taskYIELD();
  }
  vTaskDelay( 1000 / portTICK_PERIOD_MS );

}
```