Precaution against corona:

Smart soap pump bottle

Realized by:

RAJI ZINEB

April 20, 2020

Needed Components :

- 1X Arduino Uno
- 1X Servo Motor
- 1X HC-SR04 ultrasonic sensor

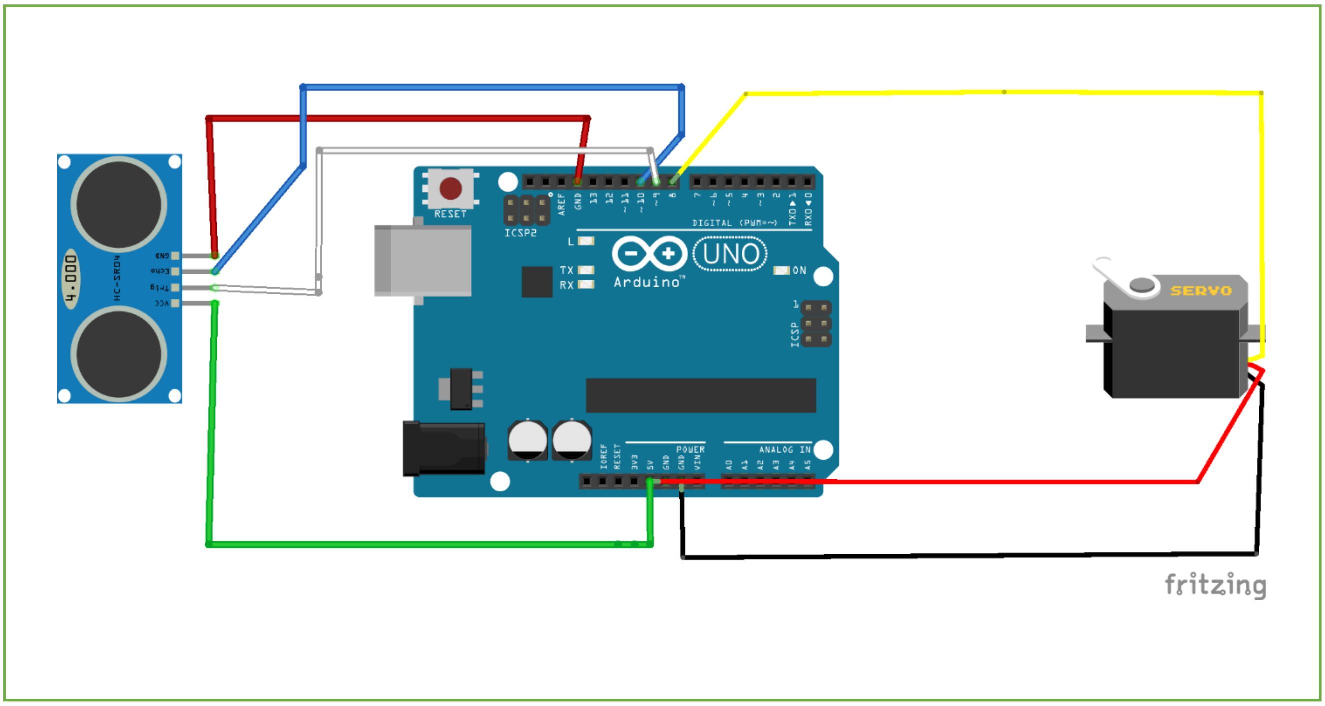
Apps and Online Service :

- Arduino Uno
- Proteus 8 Professionnel
- Fritzing

About this Project :

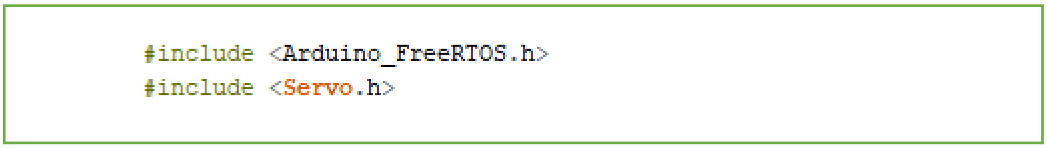
In this Project, I will make an automatic soap dispenser To make this automatic soap dispenser, I am using a ultrasonic sensor to detect motion and servomotor and the brain of this project is Arduino UNO.

Schematic :



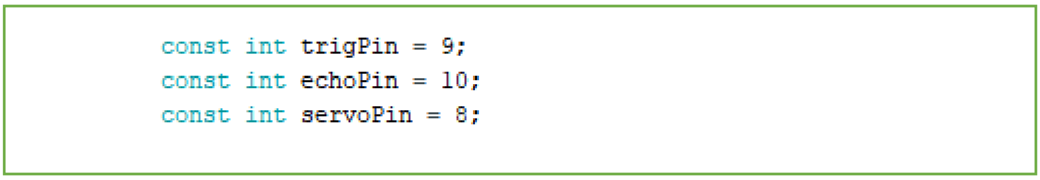
Code :

Importing Library :



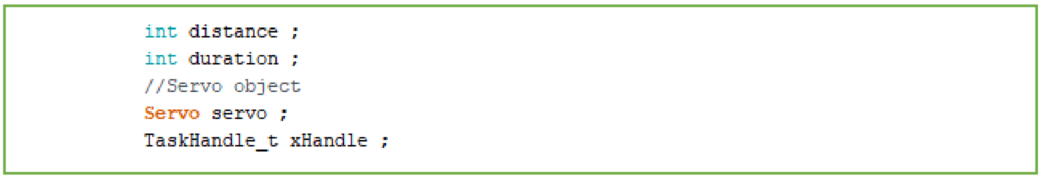
```
#include <Arduino_FreeRTOS.h>
#include <Servo.h>
```

Pins definitions for the Arduino :



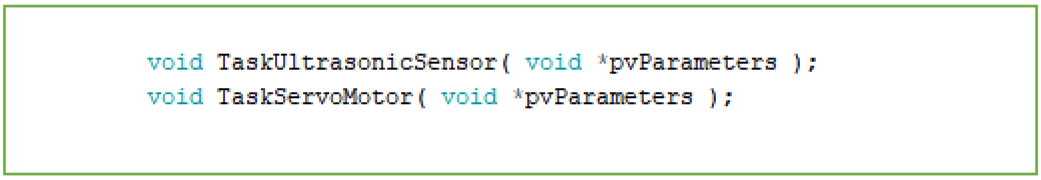
```
const int trigPin = 9;
const int echoPin = 10;
const int servoPin = 8;
```

Definition of global variables :



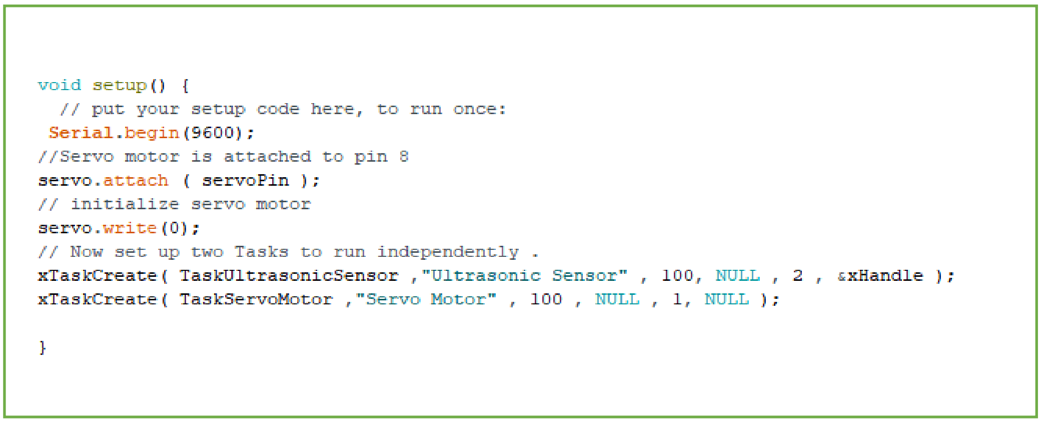
```
int distance ;
int duration ;
//Servo object
Servo servo ;
TaskHandle_t xHandle ;
```

Prototype of the tasks :



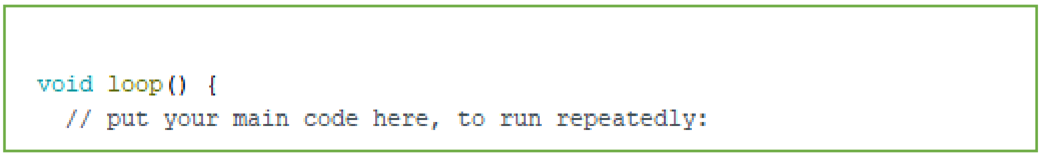
```
void TaskUltrasonicSensor( void *pvParameters );
void TaskServoMotor( void *pvParameters );
```

Setup : put your setup code here, to run once



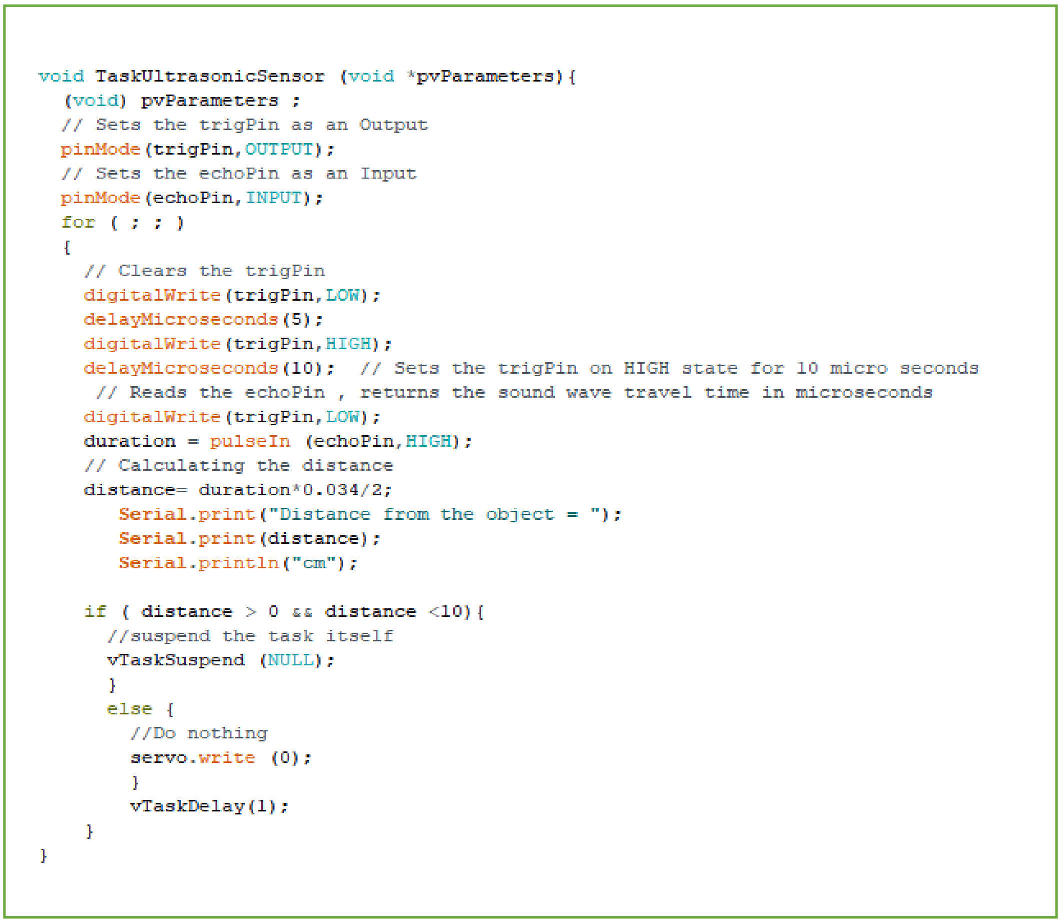
```
void setup() {
  // put your setup code here, to run once:
 Serial.begin(9600);
//Servo motor is attached to pin 8
servo.attach ( servoPin );
// initialize servo motor
servo.write(0);
// Now set up two Tasks to run independently .
xTaskCreate( TaskUltrasonicSensor ,"Ultrasonic Sensor" , 100, NULL , 2 , &xHandle );
xTaskCreate( TaskServoMotor ,"Servo Motor" , 100 , NULL , 1, NULL );

}
```

Loop :



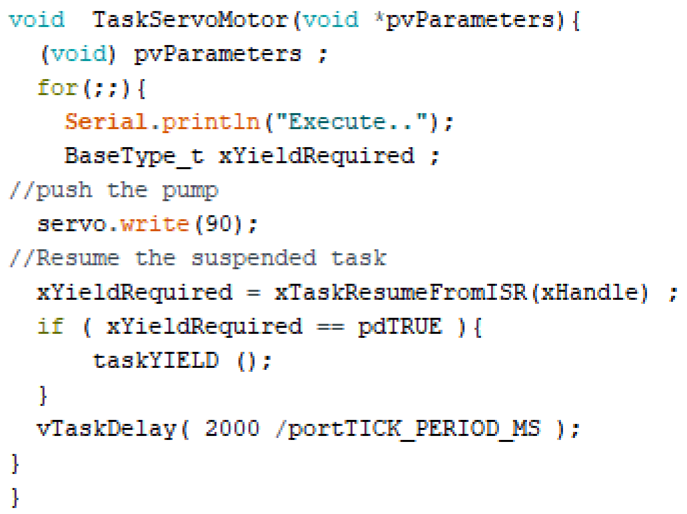
```
void loop() {
  // put your main code here, to run repeatedly:
```

Task definition : TaskUltrasonicSensor



```
void TaskUltrasonicSensor (void *pvParameters){
  (void) pvParameters ;
  // Sets the trigPin as an Output
  pinMode(trigPin,OUTPUT);
  // Sets the echoPin as an Input
  pinMode(echoPin,INPUT);
  for ( ; ; )
  {
    // Clears the trigPin
    digitalWrite(trigPin,LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin,HIGH);
    delayMicroseconds(10);  // Sets the trigPin on HIGH state for 10 micro seconds
     // Reads the echoPin , returns the sound wave travel time in microseconds
    digitalWrite(trigPin,LOW);
    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
        servo.write (0);
        }
        vTaskDelay(1);
    }
}
```

Task definition : TaskServoMotor



```
void  TaskServoMotor(void *pvParameters){
  (void) pvParameters ;
  for(;;){
    Serial.println("Execute..");
    BaseType_t xYieldRequired ;
//push the pump
  servo.write(90);
//Resume the suspended task
  xYieldRequired = xTaskResumeFromISR(xHandle) ;
  if ( xYieldRequired == pdTRUE ){
      taskYIELD ();
  }
  vTaskDelay( 2000 /portTICK_PERIOD_MS );
}
}
```