SMART WHITE CANE FOR VISUALLY IMPAIRED PEOPLE

MICROPROCESSOR SYSTEMS

Introduction

The rapid advancement in technology allows us to create devices in order to help out those who are in need, which in this particular case are visually impaired people. This device allows the user to gain a sense of their close surroundings by signaling the user if there is an obstacle in front or on the user's left or right side. One of three ultrasonic sensors is placed in the front of the device which signals the user if there is an obstacle in front by vibration and sound, or just by vibration. The other two ultrasonic sensors are placed on the right and left side of the device. Their signal determines whether an indicator attached to a servo motor moves to the left or right side of the user's hand. The device is powered by an external power supply which is connected to the microcontroller by an electrical switch, which allows the user to control when the device is turned on or off, allowing for a more economical device.

Components

- Arduino Uno Microcontroller
- Ultrasonic Sensor HC-SR04 – ultrasonic proximity sensor
- Micro Servo Motor – indicates if there is an obstacle on the right or left side of the user
- Vibration Motor – indicates if there is an obstacle in front of the user
- Buzzer (Speaker) – a sound indicator that is used in addition to the vibration motor when an obstacle is in front of the user
- Electrical Relay – auxiliary component that is used to connect two components to a single output pin
- Power source
- Electrical Switch –an on/off switch for the device

Authors: Andrej Nasteski, Monika Dimitrova

System architecture

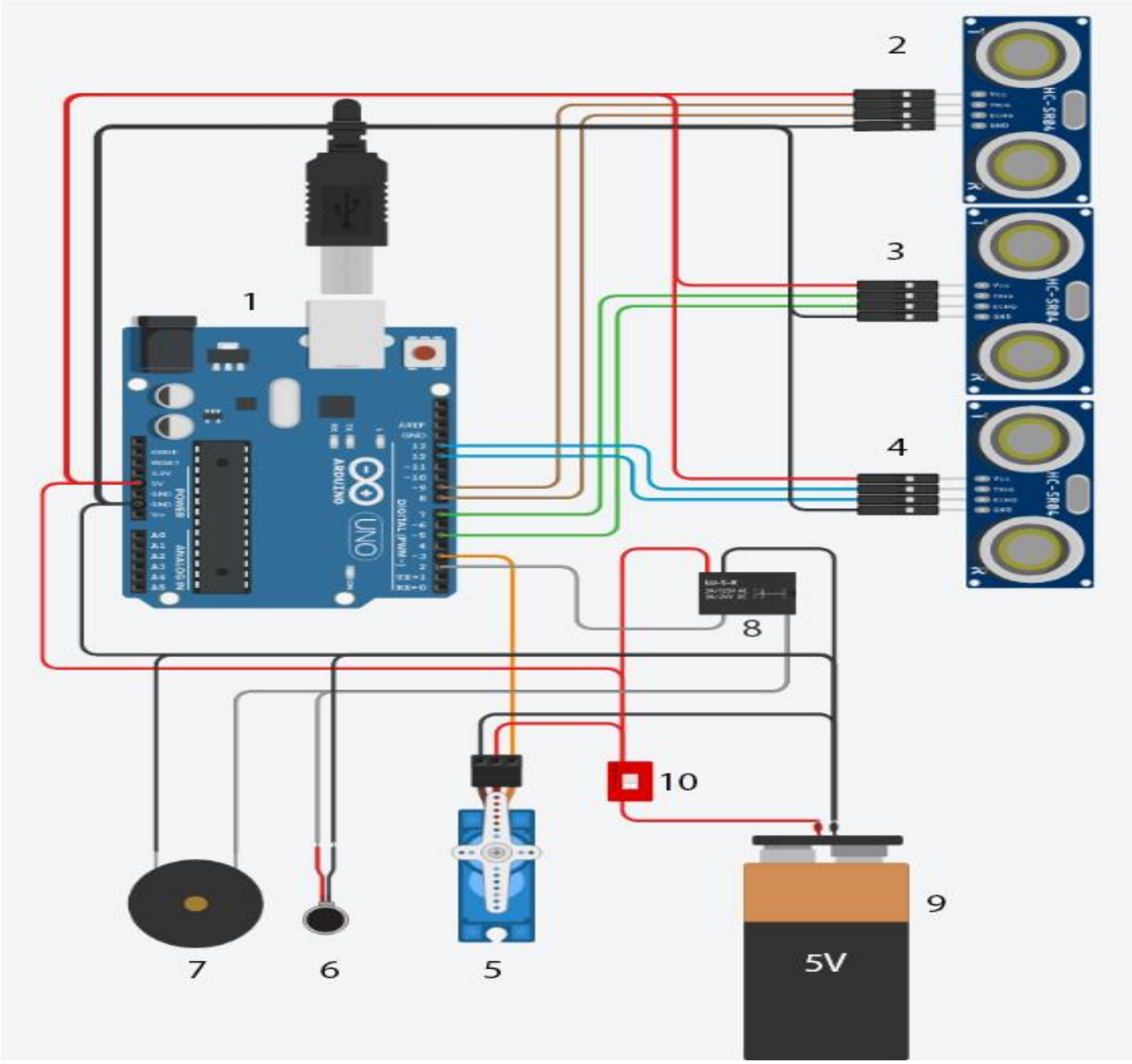


Figure 1. Component scheme

The components labeled in figure 1 are the following:

1. Arduino Uno Microcontroller
2. Ultrasound sensor – Left
3. Ultrasound sensor – Front
4. Ultrasound sensor – Right
5. Micro servo motor
6. Vibration motor
7. Sound buzzer
8. Electrical relay
9. Power source
10. Electrical switch

Authors: Andrej Nasteski, Monika Dimitrova

Prototype

We can see the full appearance of the device in figure 2, where the three ultrasound sensors are fixed in the appropriate position (left, front and right). The device is configured such that two of the three sensors are positioned to check for obstacles on the left or right side of the user. Depending on which of these two sensors is activated, a return signal is sent to the appropriate input pin, sequentially the microcontroller sends an output signal to the servo motor, positioning the indicator either to the left or right. The ultrasound sensors are configured to send a signal if an obstacle is less then 80cm away from the device (this can be configured).

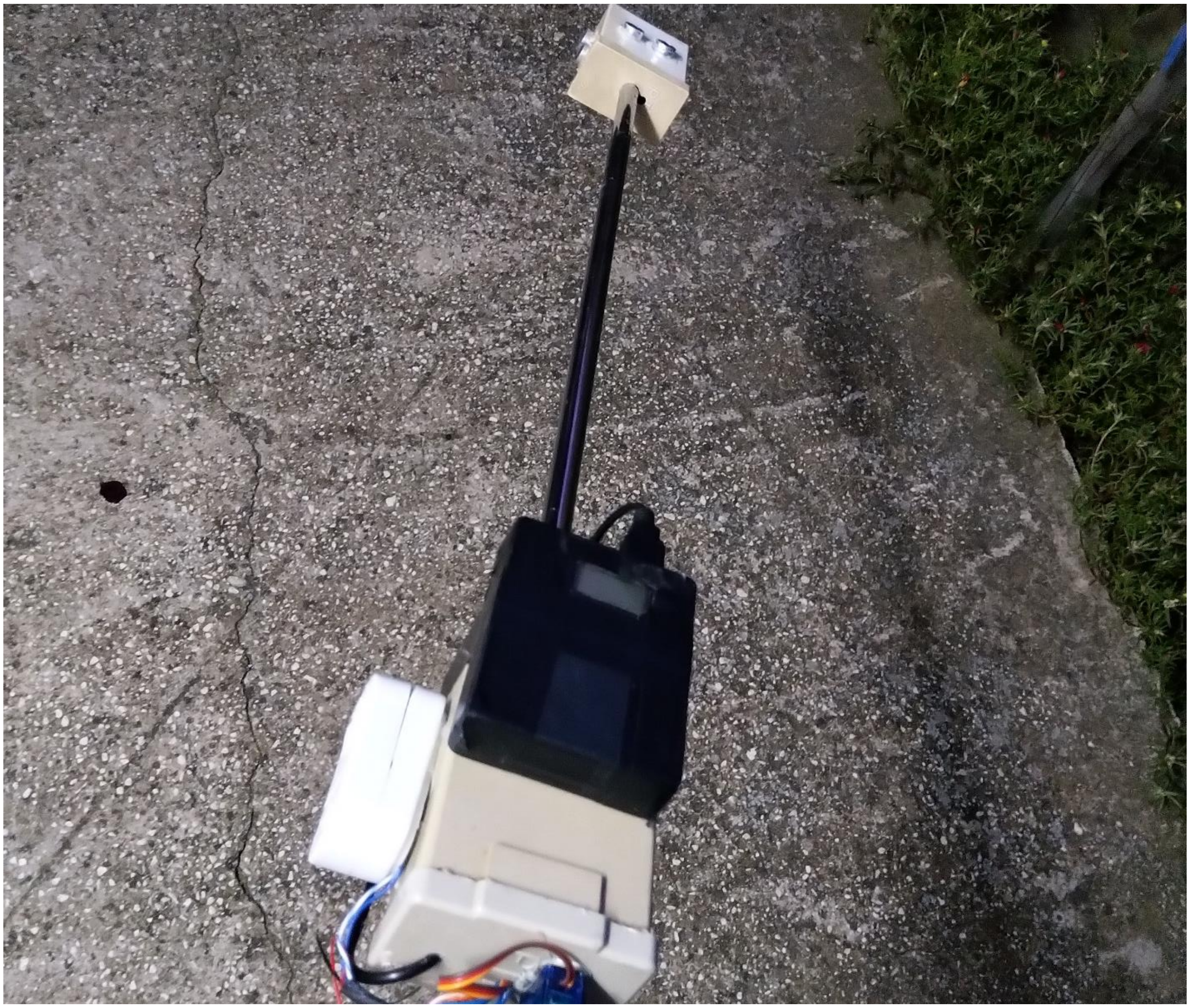


Figure 2. The assembled device.

If the front ultrasound sensor activates, a signal is sent from the microcontroller to an electrical relay, which then splits the signal to the vibration motor and the sound buzzer, activating both. Additionally, there is a switch on the buzzer that can be used to turn off the sound indication, meaning the user will be notified of an obstacle only by the vibration motor.

The microcontroller and its components are connected to an external power supply. An electrical switch is added to the device's power supply, which enables the user to turn the device on only when needed.

Authors: Andrej Nasteski, Monika Dimitrova

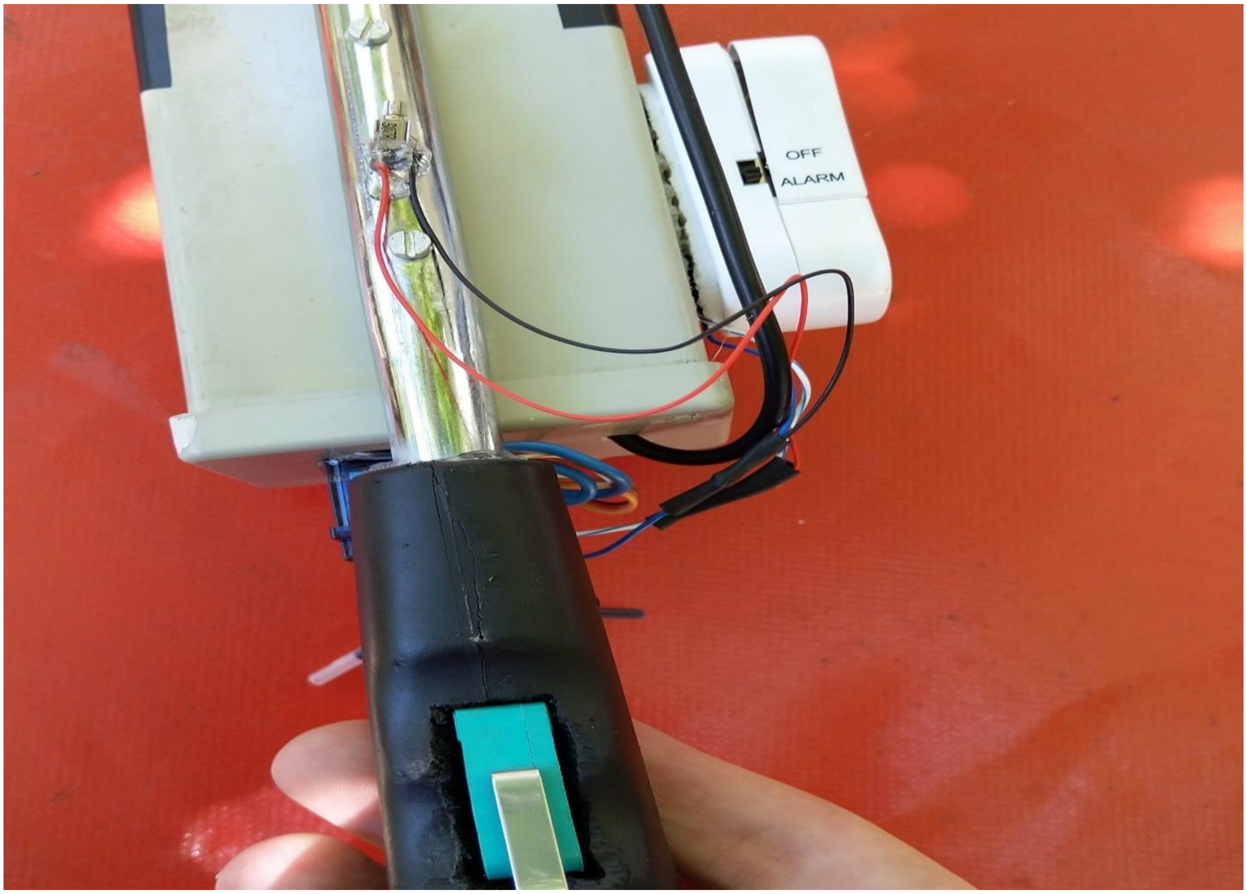


Figure 3. The electrical power switch

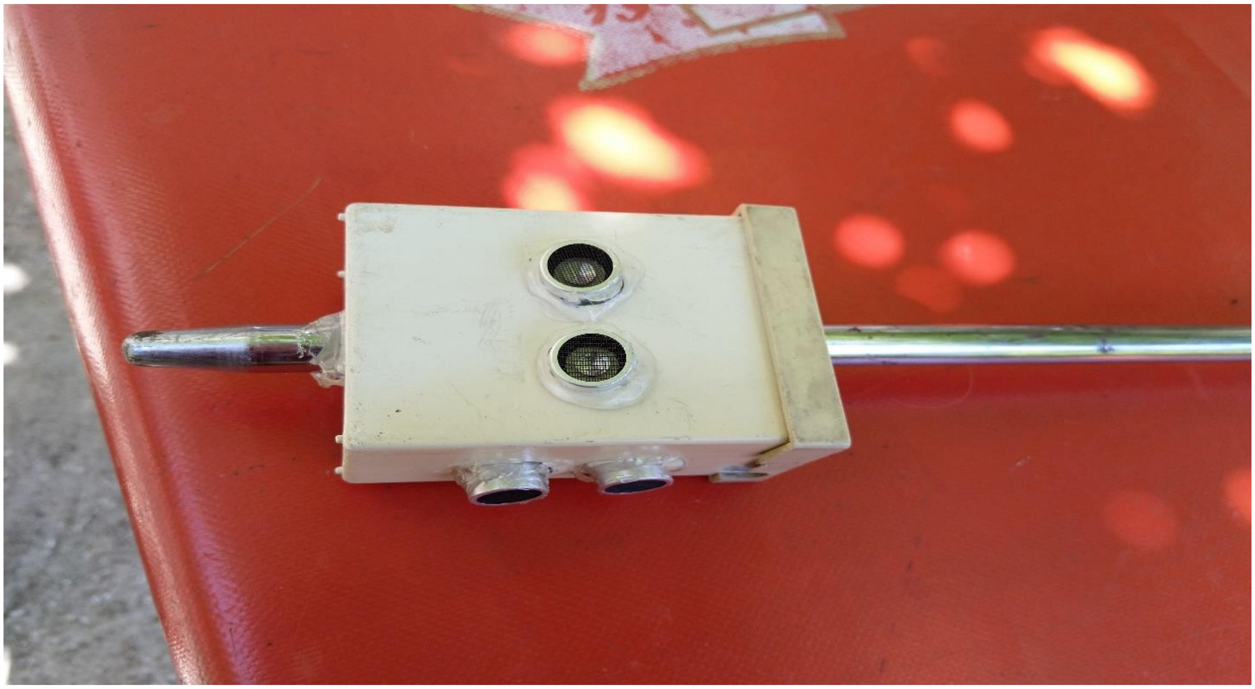
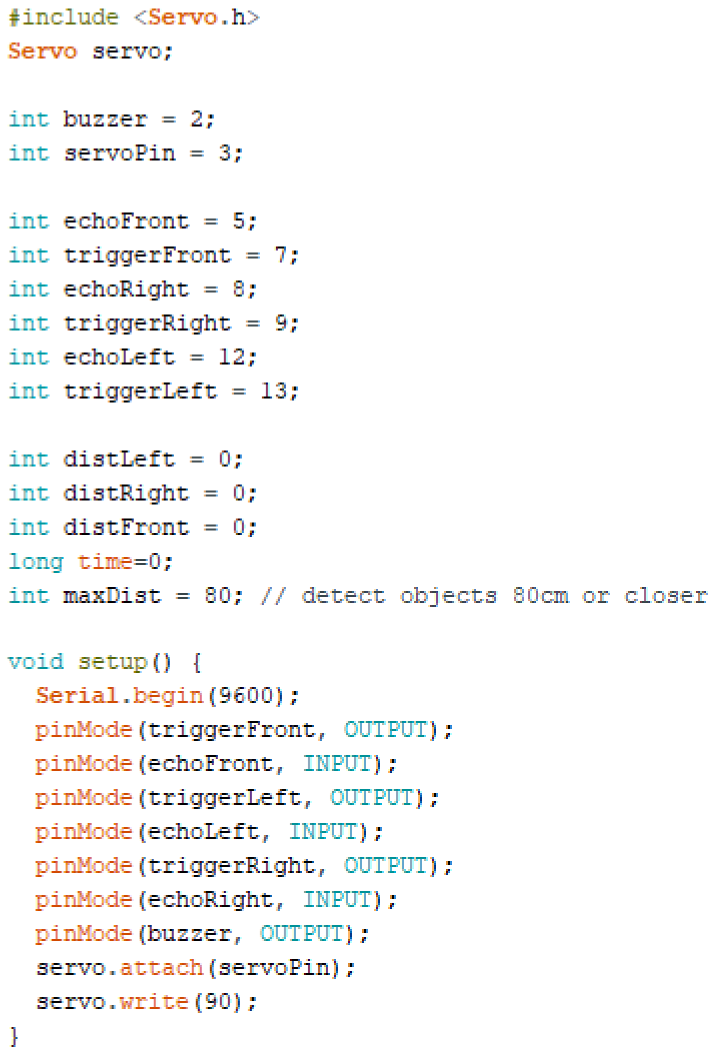


Figure 4. Ultrasonic Sensors

Authors: Andrej Nasteski, Monika Dimitrova

Algorithm



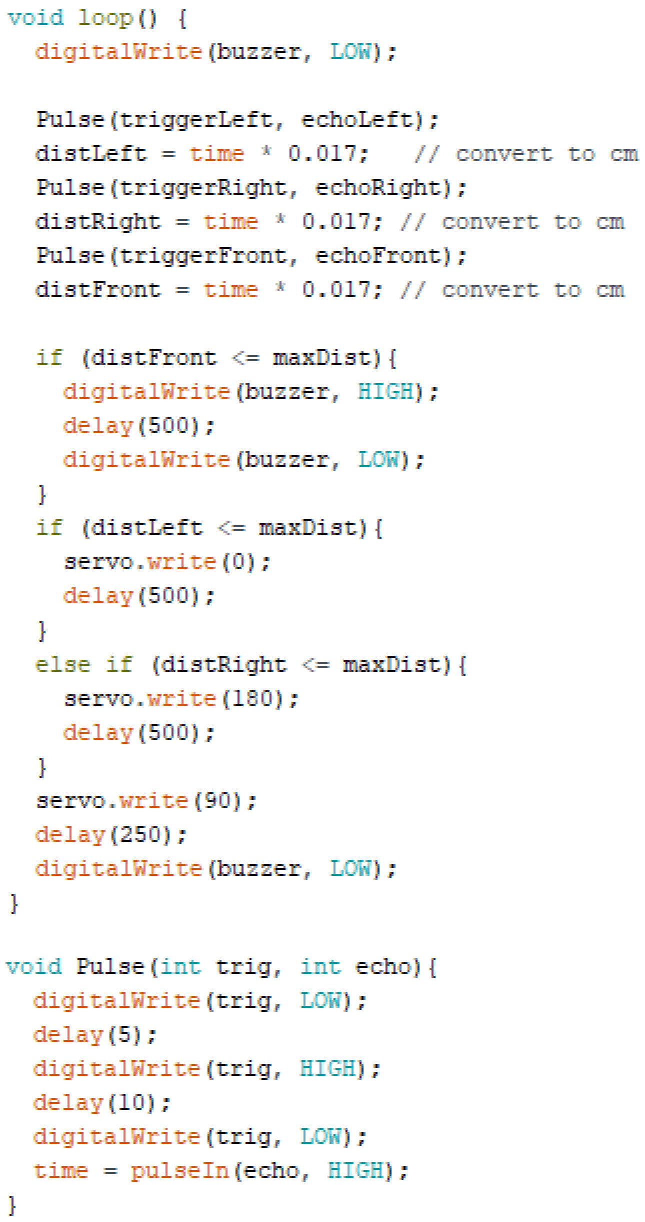
```
#include <Servo.h>
Servo servo;

int buzzer = 2;
int servoPin = 3;

int echoFront = 5;
int triggerFront = 7;
int echoRight = 8;
int triggerRight = 9;
int echoLeft = 12;
int triggerLeft = 13;

int distLeft = 0;
int distRight = 0;
int distFront = 0;
long time=0;
int maxDist = 80; // detect objects 80cm or closer

void setup() {
  Serial.begin(9600);
  pinMode(triggerFront, OUTPUT);
  pinMode(echoFront, INPUT);
  pinMode(triggerLeft, OUTPUT);
  pinMode(echoLeft, INPUT);
  pinMode(triggerRight, OUTPUT);
  pinMode(echoRight, INPUT);
  pinMode(buzzer, OUTPUT);
  servo.attach(servoPin);
  servo.write(90);
}
```



```
void loop() {
  digitalWrite(buzzer, LOW);

  Pulse(triggerLeft, echoLeft);
  distLeft = time * 0.017;    // convert to cm
  Pulse(triggerRight, echoRight);
  distRight = time * 0.017; // convert to cm
  Pulse(triggerFront, echoFront);
  distFront = time * 0.017; // convert to cm

  if (distFront <= maxDist){
    digitalWrite(buzzer, HIGH);
    delay(500);
    digitalWrite(buzzer, LOW);
  }
  if (distLeft <= maxDist){
    servo.write(0);
    delay(500);
  }
  else if (distRight <= maxDist){
    servo.write(180);
    delay(500);
  }
  servo.write(90);
  delay(250);
  digitalWrite(buzzer, LOW);
}

void Pulse(int trig, int echo){
  digitalWrite(trig, LOW);
  delay(5);
  digitalWrite(trig, HIGH);
  delay(10);
  digitalWrite(trig, LOW);
  time = pulseIn(echo, HIGH);
}
```