A Drowsy Driver Detection System

Final Term Project
Advanced Mechatronics
ME-GY 6933

Kübra Akbas and Gabrielle Cord-Cruz

Agenda

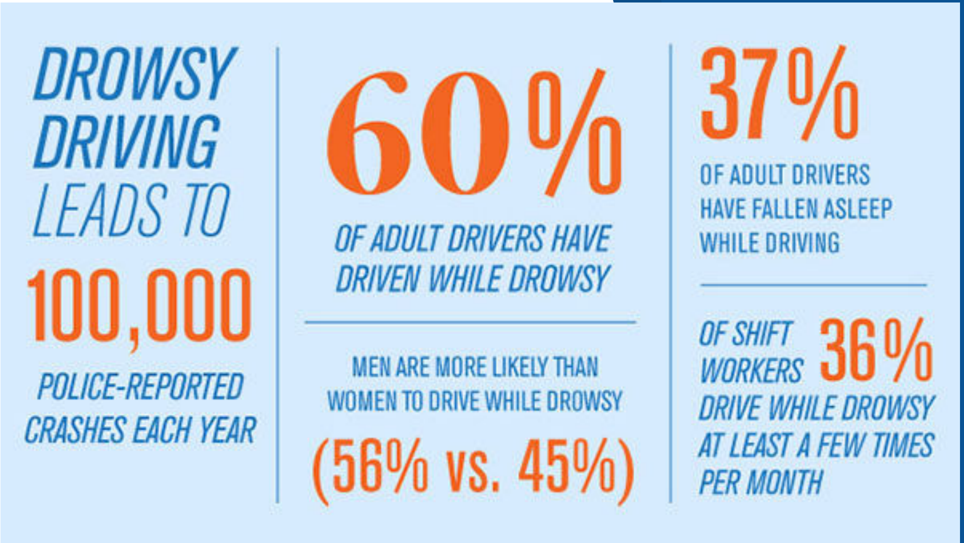
The Problem
The Solution
Design
Bill of Materials
Wiring
Code
Improvements, Future Work, Applications
Video Demo

The Problem

- Distracted driving, which can include texting, talking, eating, falling asleep, or using the radio while driving, is responsible for a large portion of automobile-related accidents
    - United States (2015): 3,450 people were killed and 391,000 were injured

The Problem

- Drowsy driving, or the practice of being fatigued or falling asleep at the wheel, contributes to a large portion of these accidents
  - United States (2013): 72,000 crashes, 44,000 injuries, and 800 deaths were attributed to drowsy driving



DROWSY DRIVING LEADS TO 100,000 POLICE-REPORTED CRASHES EACH YEAR

60% OF ADULT DRIVERS HAVE DRIVEN WHILE DROWSY

MEN ARE MORE LIKELY THAN WOMEN TO DRIVE WHILE DROWSY (56% VS. 45%)

37% OF ADULT DRIVERS HAVE FALLEN ASLEEP WHILE DRIVING

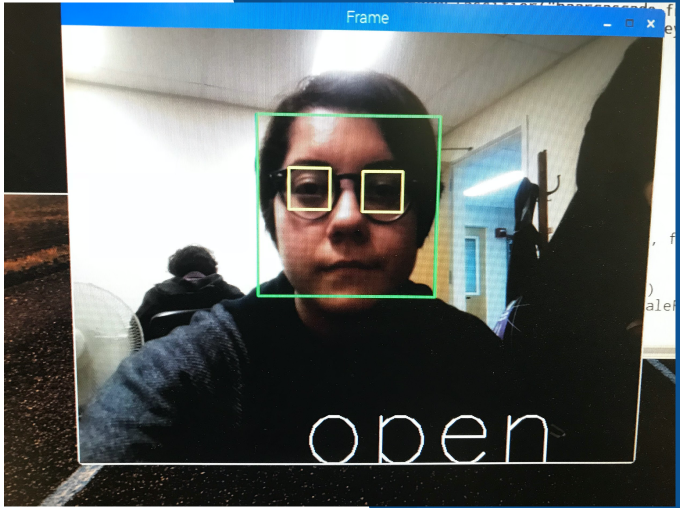
OF SHIFT WORKERS 36% DRIVE WHILE DROWSY AT LEAST A FEW TIMES PER MONTH

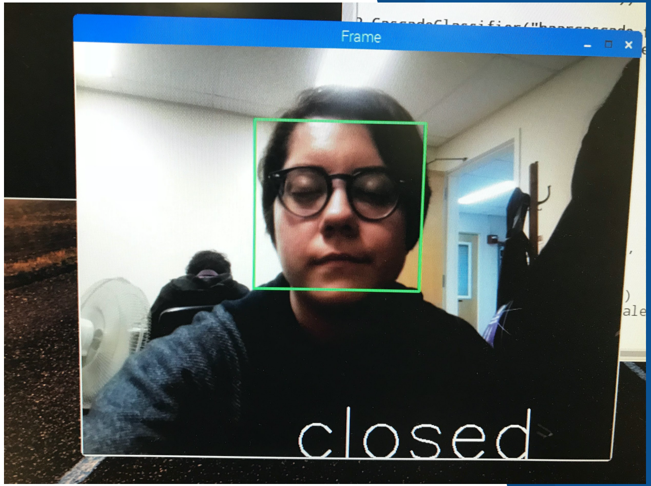
The Solution

- Create a closed-loop system to combat drowsy driving that is composed of two parts:
  - Monitoring subsystem with visual detection of drowsy driving (Raspberry Pi)
  - Response subsystem, including a wearable technology component, that alerts driver to “wake” them up or to “refocus” on the road (Arduino)
  - Serial communication between Raspberry Pi and Arduino via USB

Design

- Monitoring Subsystem:
  - Using the Pi Camera and opencv, the driver’s face and eyes are constantly monitored
- Response Subsystem:
  - Activated when eyes are closed for 3+ seconds
  - A vibration motor is mounted onto a Wii Steering Wheel
  - A smaller vibration motor is integrated into a bracelet/armband that will be worn while driving
  - An alarm is sounded on a piezo speaker

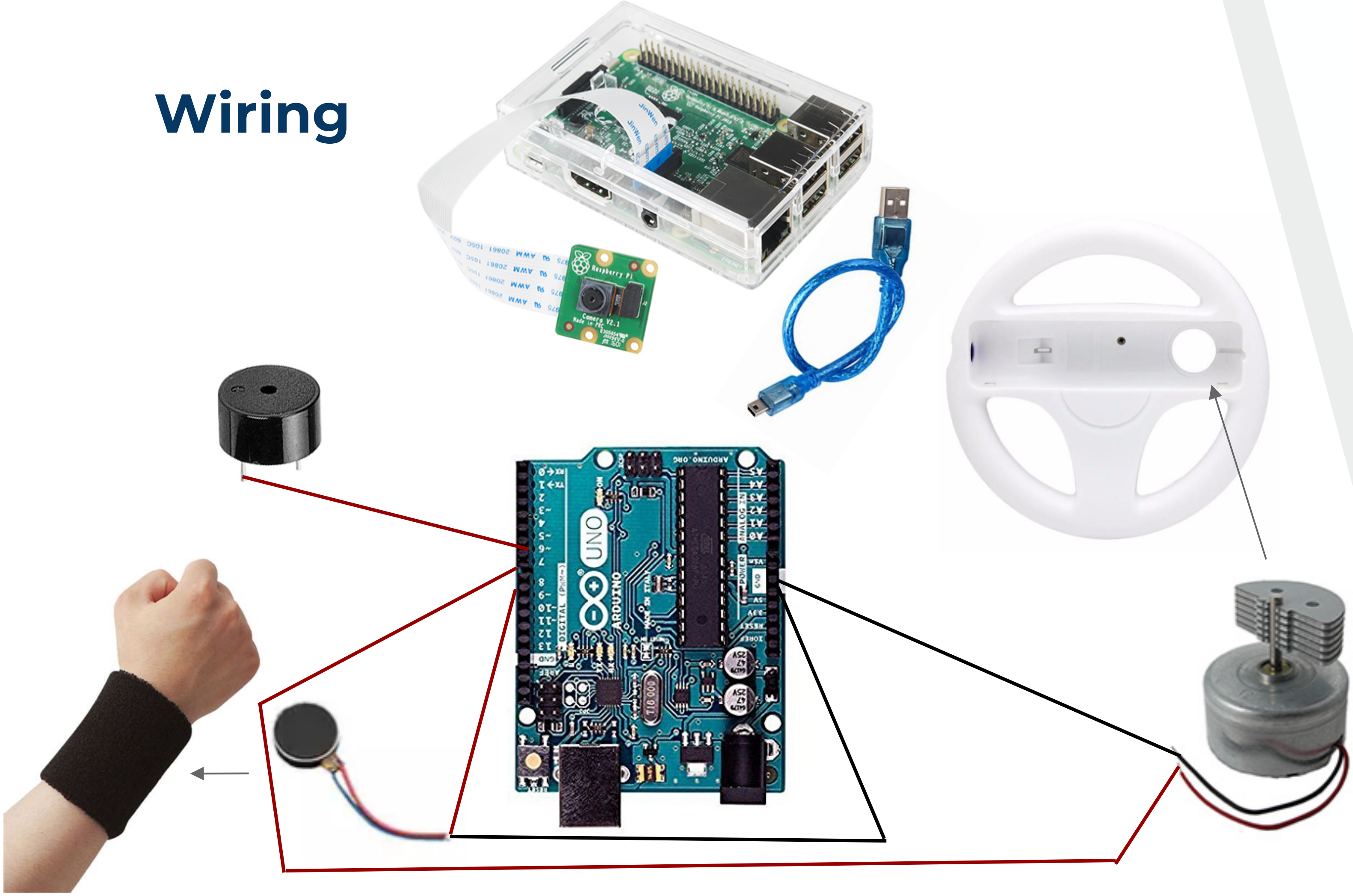




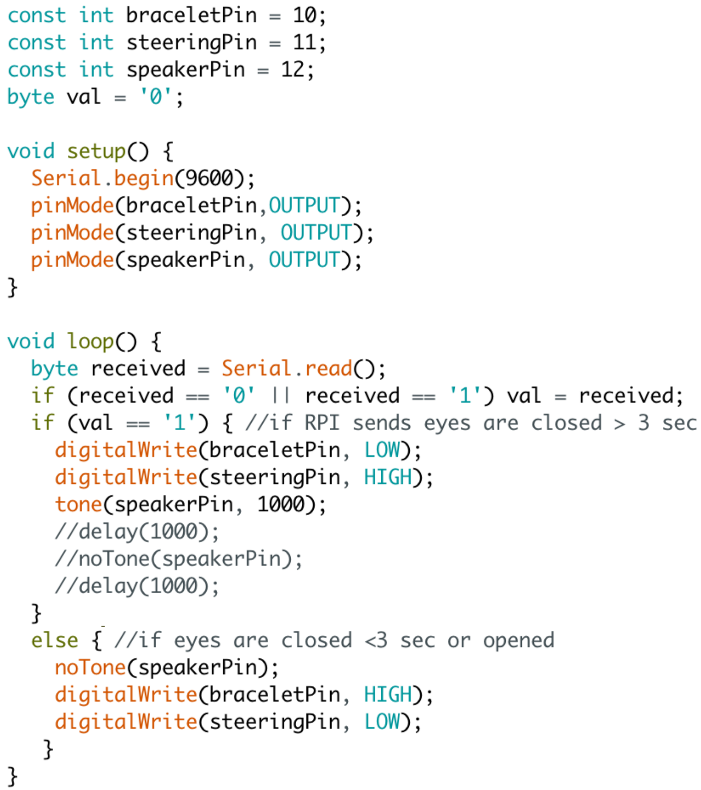
Bill of Material

Item	Quantity	Cost per Item	Cost
Raspberry Pi	1	$35.00	$35.00
Pi Camera	1	$27.88	$27.88
Wii Steering Wheel	1	$29.99	$29.99
Arduino	1	$19.99	$19.99
5V Vibration Motor	1	$5.46	$5.46
3V Vibration Motor	1	$1.22	$1.22
Micro SD Card	1	$12.00	$12.00
Velcro	1	$3.00	$3.00
Fabric	1	$2.00	$2.00
Piezo Speaker	1	$0.00	$0.00
		Total Cost	$136.54

Wiring



Code

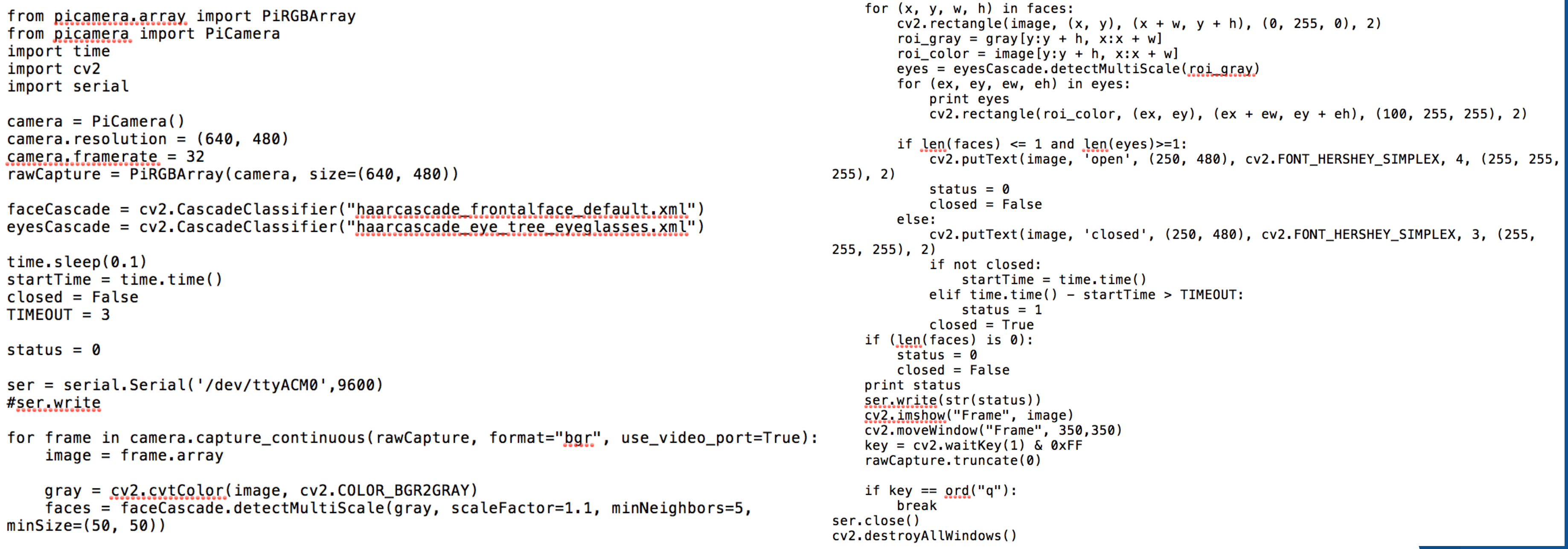


```
const int braceletPin = 10;
const int steeringPin = 11;
const int speakerPin = 12;
byte val = '0';

void setup() {
  Serial.begin(9600);
  pinMode(braceletPin,OUTPUT);
  pinMode(steeringPin, OUTPUT);
  pinMode(speakerPin, OUTPUT);
}

void loop() {
  byte received = Serial.read();
  if (received == '0' || received == '1') val = received;
  if (val == '1') { //if RPI sends eyes are closed > 3 sec
    digitalWrite(braceletPin, LOW);
    digitalWrite(steeringPin, HIGH);
    tone(speakerPin, 1000);
    //delay(1000);
    //noTone(speakerPin);
    //delay(1000);
  }
  else { //if eyes are closed <3 sec or opened
    noTone(speakerPin);
    digitalWrite(braceletPin, HIGH);
    digitalWrite(steeringPin, LOW);
   }
}
```

Code



```
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
import serial

camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(640, 480))

faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
eyesCascade = cv2.CascadeClassifier("haarcascade_eye_tree_eyeglasses.xml")

time.sleep(0.1)
startTime = time.time()
closed = False
TIMEOUT = 3

status = 0

ser = serial.Serial('/dev/ttyACM0',9600)
#ser.write

for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    image = frame.array

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(50, 50))
```

```
    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
        roi_gray = gray[y:y + h, x:x + w]
        roi_color = image[y:y + h, x:x + w]
        eyes = eyesCascade.detectMultiScale(roi_gray)
        for (ex, ey, ew, eh) in eyes:
            print eyes
            cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (100, 255, 255), 2)

        if len(faces) <= 1 and len(eyes)>=1:
            cv2.putText(image, 'open', (250, 480), cv2.FONT_HERSHEY_SIMPLEX, 4, (255, 255,
255), 2)
            status = 0
            closed = False
        else:
            cv2.putText(image, 'closed', (250, 480), cv2.FONT_HERSHEY_SIMPLEX, 3, (255,
255, 255), 2)
            if not closed:
                startTime = time.time()
            elif time.time() - startTime > TIMEOUT:
                status = 1
            closed = True
    if (len(faces) is 0):
        status = 0
        closed = False
    print status
    ser.write(str(status))
    cv2.imshow("Frame", image)
    cv2.moveWindow("Frame", 350,350)
    key = cv2.waitKey(1) & 0xFF
    rawCapture.truncate(0)

    if key == ord("q"):
        break
ser.close()
cv2.destroyAllWindows()
```

Improvements, Future Work, Applications

- Wearable technology and eye detection can be integrated into vehicles as a standard safety feature
- Integrate the alarm into the car’s stereo system
- Integrate armband with car ignition system such that the car won’t start if the armband is not worn
- Mount the Pi Camera / RPI on the dashboard or sun visor of car
- Create a fully wireless system (with a wireless bracelet)

Video Demo

Thank you!
Questions?