Parkingplace_group_9

Michiel Janssens, Viktor Nagels, Emil Dudayev, Bent Melis, Vincent Somers

LID VAN ASSOCIATIE KU LEUVEN

THOMAS MORE

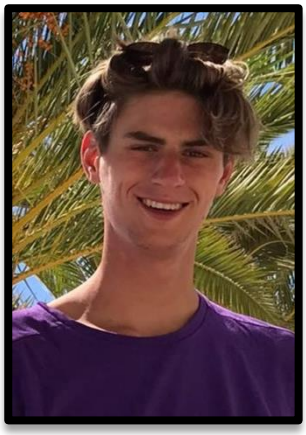
Contents

Preface

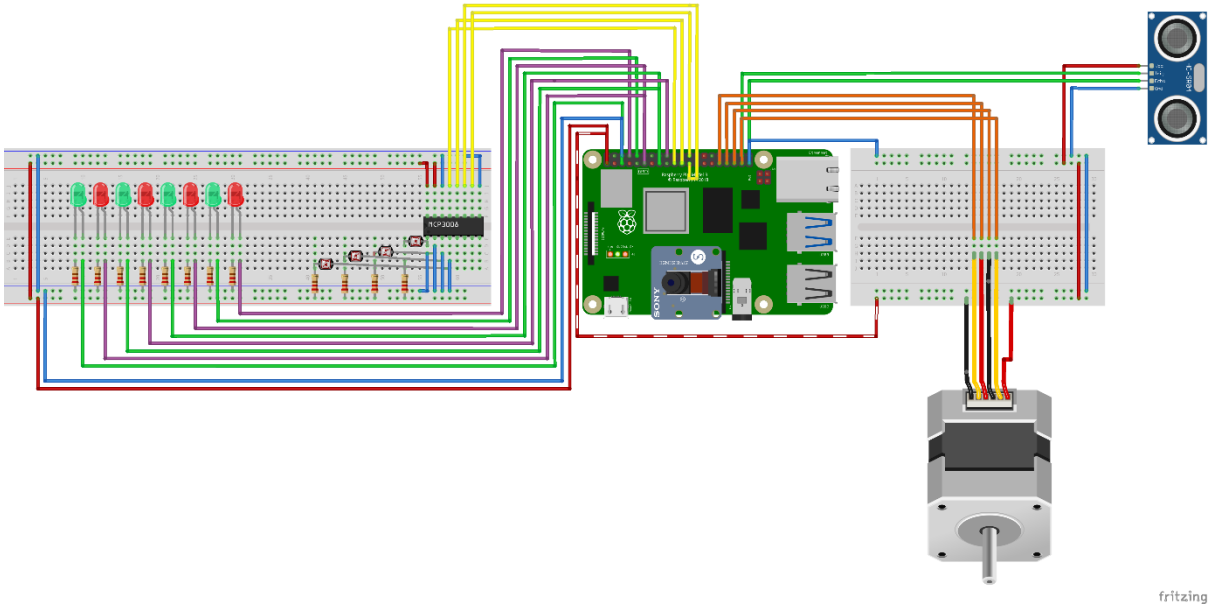
The objective of this project was to automate a parking with number plate recognition and visualization it for the users and owner. We did this on a small scale using a raspberry pi and a small scale car park.

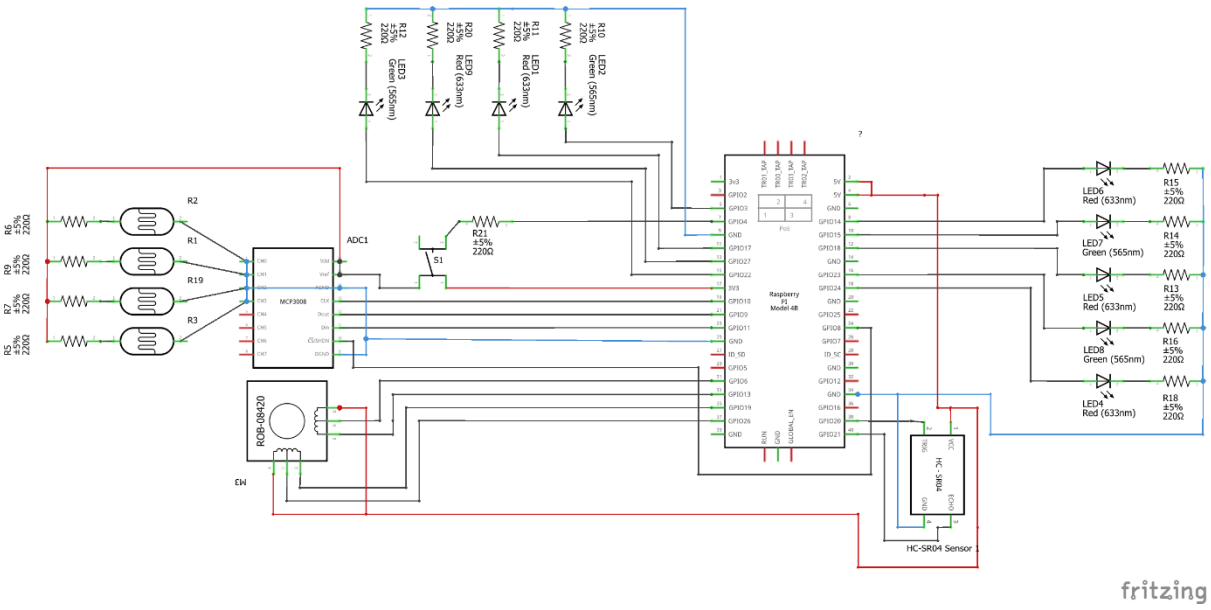
Team

Our team exist out of 5 members, each one had their own task but off course a bit of teamwork was necessary. Emil Dudayev helped with the recognition of the numberplate and helped with the documentation. Viktor was responsible for the recognition of the numberplate and also helped making the documentation. Vincent Somers was responsible for the Correct visualization on display with data from the database. Bent Melis was responsible for the correct working remote web interface and helped with the push message. And at last Michiel Janssens was responsible for the code for the barrier with ultrasonic sensor and helped with the push message.

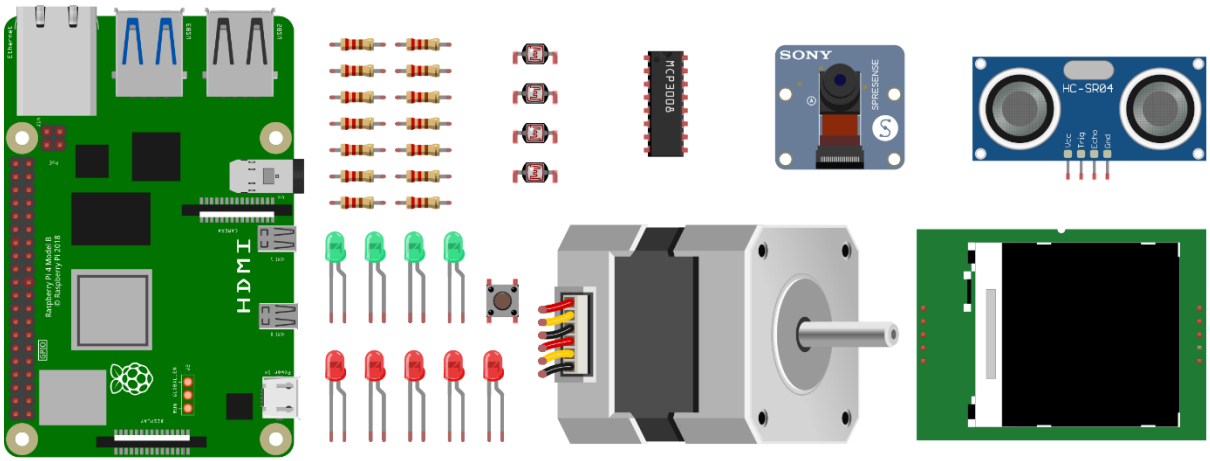


Electrical


MCP3008
SONY
HC-SR04
fritzing


R12 ±5% 220Ω
R20 ±5% 220Ω
R11 ±5% 220Ω
R10 ±5% 220Ω
LED3 Green (565nm)
LED9 Red (633nm)
LED1 Red (633nm)
LED2 Green (565nm)
R6 ±5% 220Ω
R9 ±5% 220Ω
R7 ±5% 220Ω
R5 ±5% 220Ω
R2
R1
R19
R3
ADC1
MCP3008
R21 ±5% 220Ω
S1
RCWL-0516
M3
Raspberry Pi Model 4B
LED6 Red (633nm)
LED7 Green (565nm)
LED5 Red (633nm)
LED8 Green (565nm)
LED4 Red (633nm)
R15 ±5% 220Ω
R14 ±5% 220Ω
R13 ±5% 220Ω
R16 ±5% 220Ω
R18 ±5% 220Ω
HC-SR04
HC-SR04 Sensor 1
fritzing

Parts



Code

Code explained

When the car stops in front of closed barrier with ultrasonic sensor. It will take a photo of the numberplate and it will recognize it then the barrier opens when there is a free space available. On the display at the entrance it shows which of the 4 parking spaces is still free. Then after he/she parks a car it will trigger a LDR this will be set in the database. Then on a web interface u can see which of the parking spaces are available and if the 4 parking places are occupied, it will receive this information from the database and then it will send a push message.

Full code

Python code

```
from picamera import PiCamera
from time import sleep
import requests
import RPi.GPIO as GPIO
import time
import spidev
import cgitb
import mysql.connector
import RPi.GPIO as GPIO

cgitb.enable()

ultrasonic1 = 17
ultrasonic2 = 4
button = 20
step1 = 18
step2 = 23
step3 = 24
step4 = 25
pinstep = 18,23,24,25
LED_PIN_R2 = 26
LED_PIN_G2 = 12
LED_PIN_R1 = 27
LED_PIN_G1 = 22
LED_PIN_R3 = 13
LED_PIN_G3 = 16
LED_PIN_R4 = 5
LED_PIN_G4 = 6
LED_PIN_R5 = 20
light_intensety = 950

GPIO.setmode(GPIO.BCM)
GPIO.setup((ultrasonic2), GPIO.IN)
GPIO.setup((ultrasonic1,step1,step2,step3,step4,LED_PIN_G1,LED_PIN_G2,LED_PIN_
G3,LED_PIN_G4,LED_PIN_R1,LED_PIN_R2,LED_PIN_R3,LED_PIN_R4,LED_PIN_R5),
GPIO.OUT)
#GPIO.setup(button, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.output((LED_PIN_G1, LED_PIN_G2, LED_PIN_G3, LED_PIN_G4, LED_PIN_R1,
LED_PIN_R2, LED_PIN_R3, LED_PIN_R4, LED_PIN_R5), 1)

spi = spidev.SpiDev()  # create spi object
spi.open(0, 0)  # open spi port 0, device CS0 pin 24
spi.max_speed_hz = (1000000)

def readadc(adcnum):
```

```
    if ((adcnum > 7) or (adcnum < 0)):
        return -1
    r = spi.xfer2([1, (8+adcnum) << 4, 0])
    adcout = ((r[1] & 3) << 8) + r[2]
    return adcout

def updateDB(Status, ID):
    mydb = mysql.connector.connect(
    host="localhost",
    user="pi",
    password="raspberry",
    database="ParkingDB"
    )

    mycursor = mydb.cursor()

    sql = "UPDATE ParkingLot SET bezet = %s WHERE ID = %s"
    val = (Status, ID)

    mycursor.execute(sql, val)

    mydb.commit()

    print(mycursor.rowcount, "record(s) affected")

def stepdrive(pin1,pin2,pin3,pin4):
    pinnumbers = pin1,pin2,pin3,pin4
    GPIO.setup(pinnumbers, GPIO.OUT)
    GPIO.output(pinnumbers, 0)
    GPIO.output(pin1, 1)
    GPIO.output(pin2, 1)
    GPIO.output(pin3, 0)
    GPIO.output(pin4, 0)
    time.sleep(0.01)
    GPIO.output(pin1, 0)
    GPIO.output(pin2, 1)
    GPIO.output(pin3, 1)
    GPIO.output(pin4, 0)
    time.sleep(0.01)
    GPIO.output(pin1, 0)
    GPIO.output(pin2, 0)
    GPIO.output(pin3, 1)
    GPIO.output(pin4, 1)
    time.sleep(0.01)
    GPIO.output(pin1, 1)
    GPIO.output(pin2, 0)
    GPIO.output(pin3, 0)
    GPIO.output(pin4, 1)
```

```
    time.sleep(0.01)

def photo():
    camera = PiCamera()
    camera.rotation = -90
    camera.start_preview()
    sleep(2)
    camera.capture('/home/pi/images/photo.jpg')
    camera.stop_preview()
    #camera.off()
    camera.close()

def numberplate():
    regions = ['mx', 'be'] # Change to your country
    with open('/home/pi/images/photo.jpg', 'rb') as fp:
        response = requests.post(
            'https://api.platerecognizer.com/v1/plate-reader/',
            data=dict(regions=regions),  # Optional
            files=dict(upload=fp),
            headers={'Authorization': 'Token
315a9c42be797329049bf2cc52a5cb41ab960e15'})
    json_results = (response.json())
    if (response.json()['results'] == []):
        numberplate = "FALSE"
    else:
        numberplate = (json_results['results'][0]['plate'])
        print((json_results['results'][0]['plate']))

    return numberplate

def stepmotor():
    for n in range(0, 130):
            stepdrive(step1,step2,step3,step4)
    time.sleep(10)   #TODO moet nog veranderd worden naar als de auto weg is

    for n in range(0, 130):
            stepdrive(step4,step3,step2,step1)

def ultrasonic():
    GPIO.output(ultrasonic1,1)
    time.sleep(0.00001)
    GPIO.output(ultrasonic1,0)

    while(GPIO.input(ultrasonic2)==0):
        pass
    signaalhigh = time.time()

    while(GPIO.input(ultrasonic2)==1):
```

```
        pass

    signaallow = time.time()
    timepassed = signaallow - signaalhigh
    distance = timepassed * 17000
    return distance

while True:
    buttonState = GPIO.input(button)
    distance = ultrasonic()
    print(distance)

    if distance <= 30 :    #in centimeter
        photo()
        sleep(5)
        numberplate()
        plate = numberplate()
        print(plate)
        if plate != "FALSE":
            print("Car can access parking")

            stepmotor()
        else:
            GPIO.output(LED_PIN_R5, 0)
            print("There was no numberplate found")
            time.sleep(2)
            GPIO.output(LED_PIN_R5, 1)

    else:
        print('GEEN auto aan de bareel')
        time.sleep(0.5)

    # #? Code for exiting the parking
    # if buttonState == False:
    #     print("Button was pushed!")
    #     photo()
    #     time.sleep(3)
    #     numberplate()
    #     if numberplate == "FALSE":
    #         print("car can exit parking")
    #         stepmotor()
    #     else:
    #         print("Someone is tryint to bypass the system!!!")

    lightsensor1 = readadc(1)
    lightsensor2 = readadc(0)
    lightsensor3 = readadc(2)
    lightsensor4 = readadc(3)
```

```
    print(lightsensor1, lightsensor2, lightsensor3, lightsensor4)

        #!Parking lot 1
    if lightsensor1 > light_intensety:
        GPIO.output(LED_PIN_R1, 0)
        GPIO.output(LED_PIN_G1, 1)
        updateDB(1,1)
    else:
        GPIO.output(LED_PIN_R1, 1)
        GPIO.output(LED_PIN_G1, 0)
        updateDB(0,1)
    #!Parking lot 2
    if lightsensor2 > light_intensety:
        GPIO.output(LED_PIN_R2, 0)
        GPIO.output(LED_PIN_G2, 1)
        updateDB(1,4)
    else:
        GPIO.output(LED_PIN_R2, 1)
        GPIO.output(LED_PIN_G2, 0)
        updateDB(0,4)
    #!Parking lot 3
    if lightsensor3 > light_intensety:
        GPIO.output(LED_PIN_R3, 0)
        GPIO.output(LED_PIN_G3, 1)
        updateDB(1,3)
    else:
        GPIO.output(LED_PIN_R3, 1)
        GPIO.output(LED_PIN_G3, 0)
        updateDB(0,3)

    #!Parking lot 4
    if lightsensor4 > light_intensety:
        GPIO.output(LED_PIN_R4, 0)
        GPIO.output(LED_PIN_G4, 1)
        updateDB(1,2)
    else:
        GPIO.output(LED_PIN_R4, 1)
        GPIO.output(LED_PIN_G4, 0)
        updateDB(0,2)
    time.sleep(1)

    if ((lightsensor1 < 950) and (lightsensor2 < 950) and (lightsensor3 < 950)
and (lightsensor4 < 950)):
        GPIO.output(LED_PIN_R5, 0)
    else:
        GPIO.output(LED_PIN_R5, 1)
```

LCD

```
// ParkingPlaceSCREEN

#include <GxEPD.h>
#define LILYGO_T5_V213
#include <boards.h>
#include <WiFi.h>
#include <HTTPClient.h>

int place1;
int place2;
int place3;
int place4;

// WiFi
WiFiClient wifiClient;
const char* ssid{"iPhone"};
const char* password{"bloempot"};
const char* serverName = "http://vincentsomers.sinners.be/post-esp-data.php";
const char *host = "https://vincentsomers.sinners.be";
String apiKeyValue = "tPmAT5Ab3j7F9";

// select the display class to use, only one, copy from GxEPD_Example
#include <GxDEPG0213BN/GxDEPG0213BN.h>    // 2.13" b/w  form DKE GROUP

#include <GxIO/GxIO_SPI/GxIO_SPI.h>
#include <GxIO/GxIO.h>

// constructor for AVR Arduino, copy from GxEPD_Example else
GxIO_Class io(SPI,  EPD_CS, EPD_DC,  EPD_RSET);
GxEPD_Class display(io, EPD_RSET, EPD_BUSY);

void setupWiFi();

void setup()
{
  Serial.begin(9600);
  display.init();
  display.eraseDisplay();
  setupWiFi();
  // comment out next line to have no or minimal Adafruit_GFX code
  //display.drawPaged(drawParking);
}

void setupWiFi()
{
  display.setTextColor(GxEPD_BLACK);
  Serial.println("setupWiFi");
  vTaskDelay(10 / portTICK_PERIOD_MS);
```

```
  // We start by connecting to a WiFi network
  Serial.println("Connecting to ");
  Serial.print(ssid);
  Serial.println();
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    vTaskDelay(200 / portTICK_PERIOD_MS);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.print(WiFi.localIP());
  Serial.println();
}

void drawParking()
{
  display.setTextColor(GxEPD_BLACK);
  // PLACE 1
  if (place1 == 1){
   display.fillRect(0, 0, 125, 62, GxEPD_BLACK);
  }
  display.drawLine(0, 62, 125, 62, GxEPD_BLACK);
  // PLACE 2
  if (place2 == 1){
    display.fillRect(0, 62, 125, 62, GxEPD_BLACK);
  }
  display.drawLine(0, 125, 125, 125, GxEPD_BLACK);
  // PLACE 3
  if (place3 == 1){
    display.fillRect(0, 125, 125, 125, GxEPD_BLACK);
  }
  display.drawLine(0, 187, 125, 187, GxEPD_BLACK);
  // PLACE 4
  if (place4 == 1){
    display.fillRect(0, 187, 125, 187, GxEPD_BLACK);
  }

  if (place1 == 1 && place2 == 1 && place3 == 1 && place4 == 1){

  display.setTextColor(GxEPD_WHITE);
  display.setCursor(25, 125);
  display.print("PARKING FULL");
  }
}
```

```
void loop() {
  display.init();
  display.eraseDisplay();
  place1 = 0;
  place2 = 0;
  place3 = 0;
  place4 = 0;
  // comment out next line to have no or minimal Adafruit_GFX code

  // put your main code here, to run repeatedly:
  HTTPClient http; //--> Declare object of class HTTPClient

  //-----------------------------------------Getting Data from MySQL Database
  String GetAddress, LinkGet, getData;
  int id = 1; //--> ID in Database
  GetAddress = "/esp-data.php";
  LinkGet = host + GetAddress; //--> Make a Specify request destination
  getData = "id=" + String(id);
  Serial.println("----------------Connect to Server----------------");
  Serial.println("Get Parking Status from Server or Database");
  Serial.print("Request Link : ");
  Serial.println(LinkGet);
  http.begin(LinkGet); //--> Specify request destination
  http.addHeader("Content-Type", "application/x-www-form-
urlencoded");    //Specify content-type header
  int httpCodeGet = http.POST(getData); //--> Send the request
  String payloadGet = http.getString(); //--> Get the response payload from
server
  Serial.print("Response Code : "); //--> If Response Code = 200 means
Successful connection, if -1 means connection failed. For more information see
here : https://en.wikipedia.org/wiki/List_of_HTTP_status_codes
  Serial.println(httpCodeGet); //--> Print HTTP return code
  Serial.print("Returned data from Server : ");
  Serial.println(payloadGet); //--> Print request response payload

  if(payloadGet.indexOf("1") >= 0){
    place1 = 1;
  }

   if(payloadGet.indexOf("2") >= 0){
    place2 = 1;
  }

   if(payloadGet.indexOf("3") >= 0){
    place3 = 1;
  }
```

```
 if(payloadGet.indexOf("4") >= 0){
  place4 = 1;
}

Serial.println("----------------Closing Connection----------------");
http.end(); //--> Close connection
Serial.println();
Serial.println("Please wait 10 seconds for the next connection.");
Serial.println();
display.drawPaged(drawParking);
delay(10000);
};
```

PHP

Index pagina

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Parking</title>
</head>
<body>
    <?php
    $hostname = "localhost";
    $username = "pi";
    $password = "raspberry";
    $db = "ParkingDB";
    $dbconnect=mysqli_connect($hostname,$username,$password,$db);
    if ($dbconnect->connect_error) {
        die("Database connection failed: " . $dbconnect->connect_error);
    }
    $query = mysqli_query($dbconnect, "SELECT bezet FROM `ParkingLot`")
        or die (mysqli_error($dbconnect));
    while ($row = mysqli_fetch_array($query)) {
        $bezet[] = $row[0];
    }
    $rood = '#E16950';
    $groen = '#91F160';
    ?>
    <h1>
        Parking webinterface
    </h1>
    <svg width="150" height="360">
        <rect id="parking1" x="0" y="0" width="150" height="360" style="fill:
        <?php
            if($bezet[0] == 0){
                echo $groen;
            }
            else{
                echo $rood;
            }
        ?>
        ;stroke:black;stroke-width:5;" />
    </svg>
    <svg width="150" height="360">
        <rect id="parking1" x="0" y="0" width="150" height="360" style="fill:
        <?php
            if($bezet[1] == 0){
                echo $groen;
```

```
            }
            else{
                echo $rood;
            }
        ?>
        ;stroke:black;stroke-width:5;" />
    </svg>
    <svg width="150" height="360">
        <rect id="parking1" x="0" y="0" width="150" height="360" style="fill:
        <?php
            if($bezet[2] == 0){
                echo $groen;
            }
            else{
                echo $rood;
            }
        ?>
        ;stroke:black;stroke-width:5;" />
    </svg>
    <svg width="150" height="360">
        <rect id="parking1" x="0" y="0" width="150" height="360" style="fill:
        <?php
            if($bezet[3] == 0){
                echo $groen;
            }
            else{
                echo $rood;
            }
        ?>
        ;stroke:black;stroke-width:5;" />
    </svg>
    <div> Aantal vrije plaatsen: <span style="font-weight:bold">
    <?php
    $counts = array_count_values($bezet);

    if($bezet['0'] == 1 and $bezet['1'] == 1 and $bezet['2'] == 1 and
$bezet['3'] == 1){
        echo '0';
    }
    else{
        echo $counts['0'];
    }
    ?></span>
    </div>
    <?php
    $counts = array_count_values($bezet);
    if ($counts['0'] == 0){
        curl_setopt_array($ch = curl_init(), array(
```

```
        CURLOPT_URL => "https://api.pushover.net/1/messages.json",
        CURLOPT_POSTFIELDS => array(
            "token" => "adcxfe2q99fibaupf4arwf3cdi7sa4",
            "user" => "uabtbvyuxaudjddqp86o5rh5xordta",
            "message" => "Youre parking is full! 🚗",
        ),
        CURLOPT_SAFE_UPLOAD => true,
        CURLOPT_RETURNTRANSFER => true,
        ));
        curl_exec($ch);
        curl_close($ch);

        $servername = "localhost";
        $username = "pi";
        $password = "raspberry";
        $dbname = "ParkingDB";

        // Create connection
        $conn = new mysqli($servername, $username, $password, $dbname);
        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

        $sql = "UPDATE ParkingLot SET bezet=0 WHERE id=5";

        if ($conn->query($sql) === TRUE) {
            echo "";
        }
        else {
            echo "";
        }
    }
    if ($bezet['0'] == 0 or $bezet['1'] == 0 or $bezet['2'] == 0 or
$bezet['3'] == 0 and $bezet[4] == 0){

        $servername = "localhost";
        $username = "pi";
        $password = "raspberry";
        $dbname = "ParkingDB";

        // Create connection
        $conn = new mysqli($servername, $username, $password, $dbname);
        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }
```

```
        $sql = "UPDATE ParkingLot SET bezet=1 WHERE id=5";

        if ($conn->query($sql) === TRUE) {
            echo "";
        }
        else {
            echo "";
        }
    }

    header("refresh: 3");
    ?>
</body>
</html>
```

ESP Data

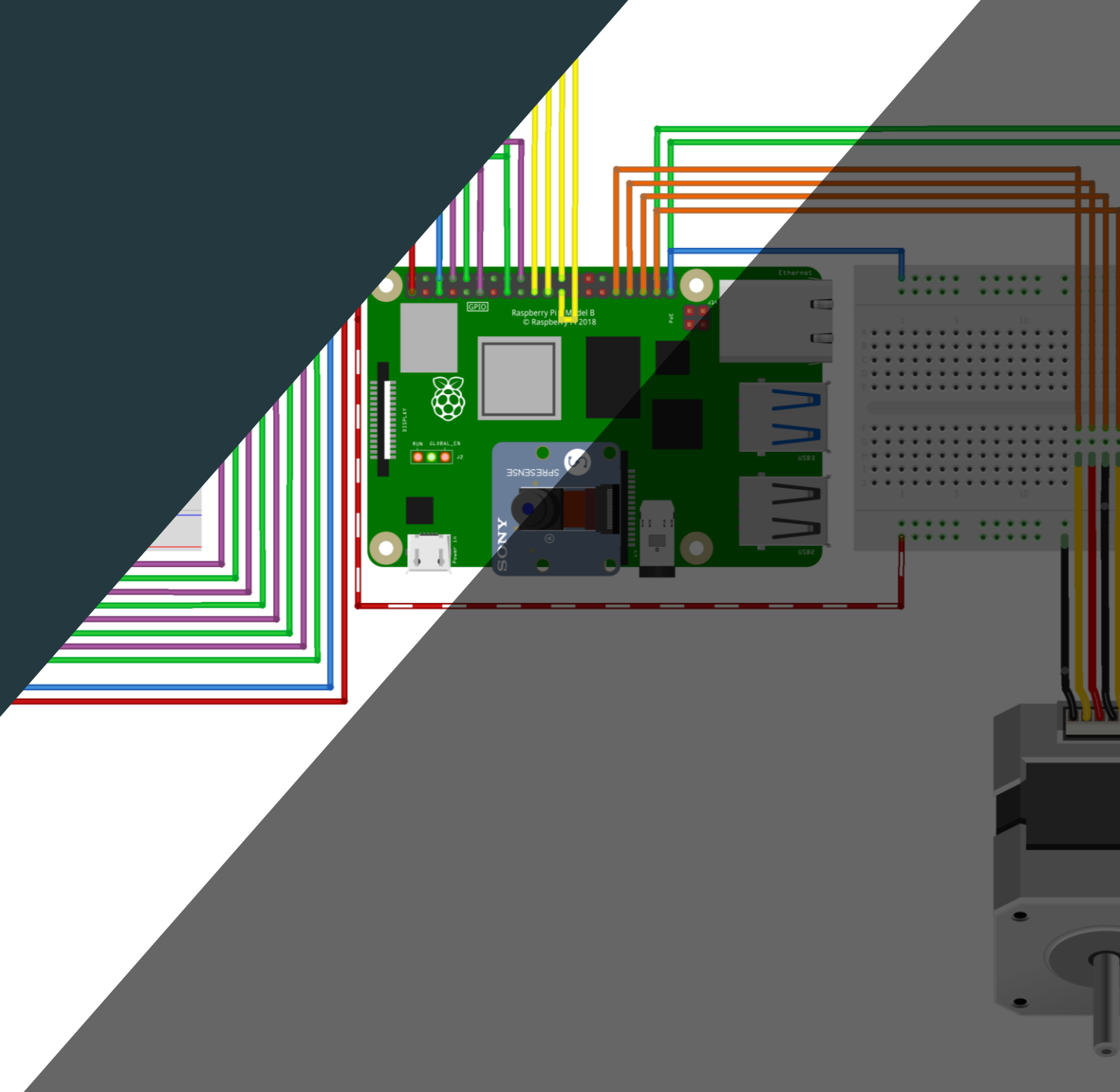
```
<?php

$servername = "db.sinners.be";

$dbname = "vincentsomers_plantdata";
$username = "vincentsomers";
$password = "cKWtZSjfterq";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id FROM display where status = 1";

if ($result = $conn->query($sql)) {
    while ($row = $result->fetch_assoc()) {
        $row_id = $row["id"];
        $row_status = $row["status"];

        echo ''. $row_id .'';
    }
    $result->free();
}
$conn->close();
?>
```

Link video

https://youtu.be/S7kWJHW7ot8

PARKING PLACE

Michiel Janssens, Emil Dudayev, Viktor Nagels, Bent Melis, Vincent Somers



team





Plan of approach

Demo