

Parkingplace\_group\_9

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

Contents

[Preface 3](#_Toc89291567)

[Team 3](#_Toc89291568)

[Electrical 4](#_Toc89291569)

[Parts 5](#_Toc89291570)

[Code 5](#_Toc89291571)

[Code explained 5](#_Toc89291572)

[Full code 6](#_Toc89291573)

[Python code 6](#_Toc89291574)

[LCD 11](#_Toc89291575)

[PHP 15](#_Toc89291576)

[Link video 19](#_Toc89291577)

# 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.

A picture containing person, person, outdoor, shirt

Description automatically generated

Afbeelding met boom, persoon, buiten, plant

Automatisch gegenereerde beschrijvingA person wearing glasses

Description automatically generated

# Electrical

Diagram

Description automatically generated with medium confidence

Diagram, schematic

Description automatically generated

# Parts

A picture containing diagram

Description automatically generated

# 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**

<!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