

Parkingplace\_group\_9

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

Contents

[Preface 3](#_Toc89270729)

[Team 3](#_Toc89270730)

[Electrical 4](#_Toc89270731)

[Parts 5](#_Toc89270732)

[Code 5](#_Toc89270733)

[Code explained 5](#_Toc89270734)

[Full code 6](#_Toc89270735)

[Python code 6](#_Toc89270736)

[LCD 9](#_Toc89270737)

[PHP 13](#_Toc89270738)

# 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

A person wearing glasses

Description automatically generatedA picture containing outdoor, person, snow, child

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

## Full code

### Python code

from picamera import PiCamera

from time import sleep

import requests

from pprint import pprint #? nakijken nog nodig

import RPi.GPIO as GPIO

import time

ultrasonic1 = 20

ultrasonic2 = 21

button = 12

step1 = 6

step2 = 13

step3 = 19

step4 = 26

GPIO.setmode(GPIO.BCM)

GPIO.setup((ultrasonic2,button), GPIO.IN)

GPIO.setup((ultrasonic1,step1,step2,step3,step4), GPIO.OUT)

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() # TODO deze rotatie kan nog aangepast worden A.D.H.V. hoe de camera geposisioneert staat

    camera.rotation = 0

    camera.start\_preview()

    sleep(2)

    camera.capture('/home/pi/images/photo.jpg')  # TODO make location for the pictures

    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'])

        pprint ((json\_results['results'][0]['plate']))

        pprint (response.json())

    return numberplate

def stepmotor():

    for n in range(0, 130):

            stepdrive(step1,step2,step3,step4)

    time.sleep(10)

    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:

    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:

            print("There was no numberplate found")

    else:

        GPIO.output(step1, 0)

        GPIO.output(step2, 0)

        GPIO.output(step3, 0)

        GPIO.output(step4, 0)

        print('GEEN auto aan de bareel')

        time.sleep(0.5)

    #? Code for exiting the parking

    if GPIO.input(button) == GPIO.HIGH:

        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!!!")

### 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>Parkinglots</title>

</head>

<body>

    <?php

    $hostname = "db.sinners.be";

    $username = "viktornagels";

    $password = "LV7ha6A2HyzR";

    $db = "viktornagels\_IoT\_Advanced\_Project";

    $dbconnect=mysqli\_connect($hostname,$username,$password,$db);

    if ($dbconnect->connect\_error) {

        die("Database connection failed: " . $dbconnect->connect\_error);

    }

    $query = mysqli\_query($dbconnect, "SELECT StatusParkingLot FROM `parkingLots`")

        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);

    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);

        $hostname = "db.sinners.be";

        $username = "viktornagels";

        $password = "LV7ha6A2HyzR";

        $db = "viktornagels\_IoT\_Advanced\_Project";

        // Create connection

        $conn = new mysqli($servername, $username, $password, $dbname);

        // Check connection

        if ($conn->connect\_error) {

            die("Connection failed test: " . $conn->connect\_error);

        }

        $sql = "UPDATE parkingLots SET StatusParkingLot=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){

        $hostname = "db.sinners.be";

        $username = "viktornagels";

        $password = "LV7ha6A2HyzR";

        $db = "viktornagels\_IoT\_Advanced\_Project";

        // Create connection

        $conn = new mysqli($servername, $username, $password, $dbname);

        // Check connection

        if ($conn->connect\_error) {

            die("Connection failed---: " . $conn->connect\_error);

        }

        $sql = "UPDATE parkingLots SET StatusParkingLot=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();

?>