

# Parking Slot Sensor

## PROJECT IIOTCA

CHIS HORIA & BADILA TIMOTEI

GR. 1.1

GR. THURSDAY 16:00

### Project Idea & Main Features

We created a "Parking Slot Sensor" using a web camera[5], which detects what parking slot is free.

Using our code and our camera, this sensor recognise letters using AI Identification.

If the sensor recognises the letter, it shows a notification that this parking slot is empty.

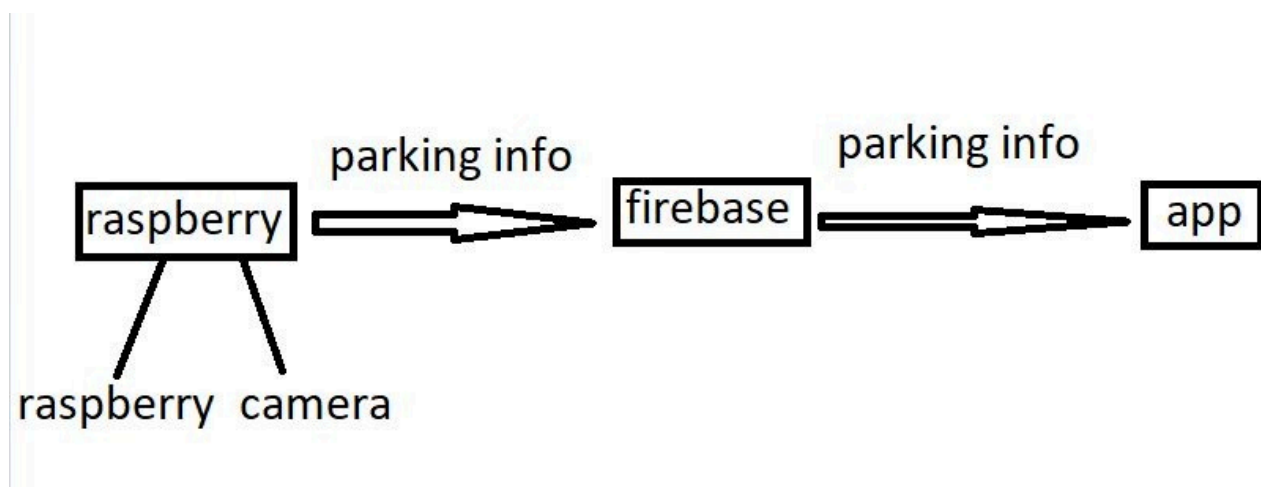
Example: "A3 parking slot is empty."

The final example/result is a notification, using a cloud database, with a list with all the empty parking slots.

"Empty parking slots: A3, A5, B2 ,B4 ,B5 are empty."

### Code

### System architecture



## Connecting our sensor to Firebase cloud database

```
config = {
    'apiKey': 'AIzaSyAbXlaAYcsovACZIAygJEo9nGEue3d4Hyw',
    'authDomain': 'parking-iiotca.firebaseio.com',
    'databaseURL': 'https://parking-iiotca-default-rtdb.firebaseio.com/',
    'storageBucket': 'parking-iiotca.appspot.com'
}

firebase = pyrebase.initialize_app(config)
db = firebase.database()
```

## Initialising the web camera

```
# initialize camera
print('Initializing camera')
camera=cv2.VideoCapture(1)

# initialize text-detector
print('Initializing cv algorithms')
reader = easyocr.Reader(['en'], gpu = False)

park_score = {}
frames_running = 0
```

## The whole process

```
while True:
    # get image from camera
    _, image = camera.read()
    cv2.imshow('Text detection', image)

    frames_running = frames_running + 1

    # process image
    if cv2.waitKey(2):
        # detect text
        results = reader.readtext(image)

        # get parking codes
        park_codes = [x[1] for x in results]

        # divide mixed parking codes
        park_codes_corr = []
        for code in park_codes:
            code.replace(' ', '')
            code.replace('S', '5')
            codes = re.split(r'[I() l\\V/]', code)
            park_codes_corr = park_codes_corr + codes

        # keep only valid parking codes
        park_codes_checked = []
        for code in park_codes_corr:
            if len(code) == 2 and code[0].isalpha() and code[1].isdigit():
                park_codes_checked.append(code)

        print(park_codes)
        print(park_codes_corr)
        print(park_codes_checked)

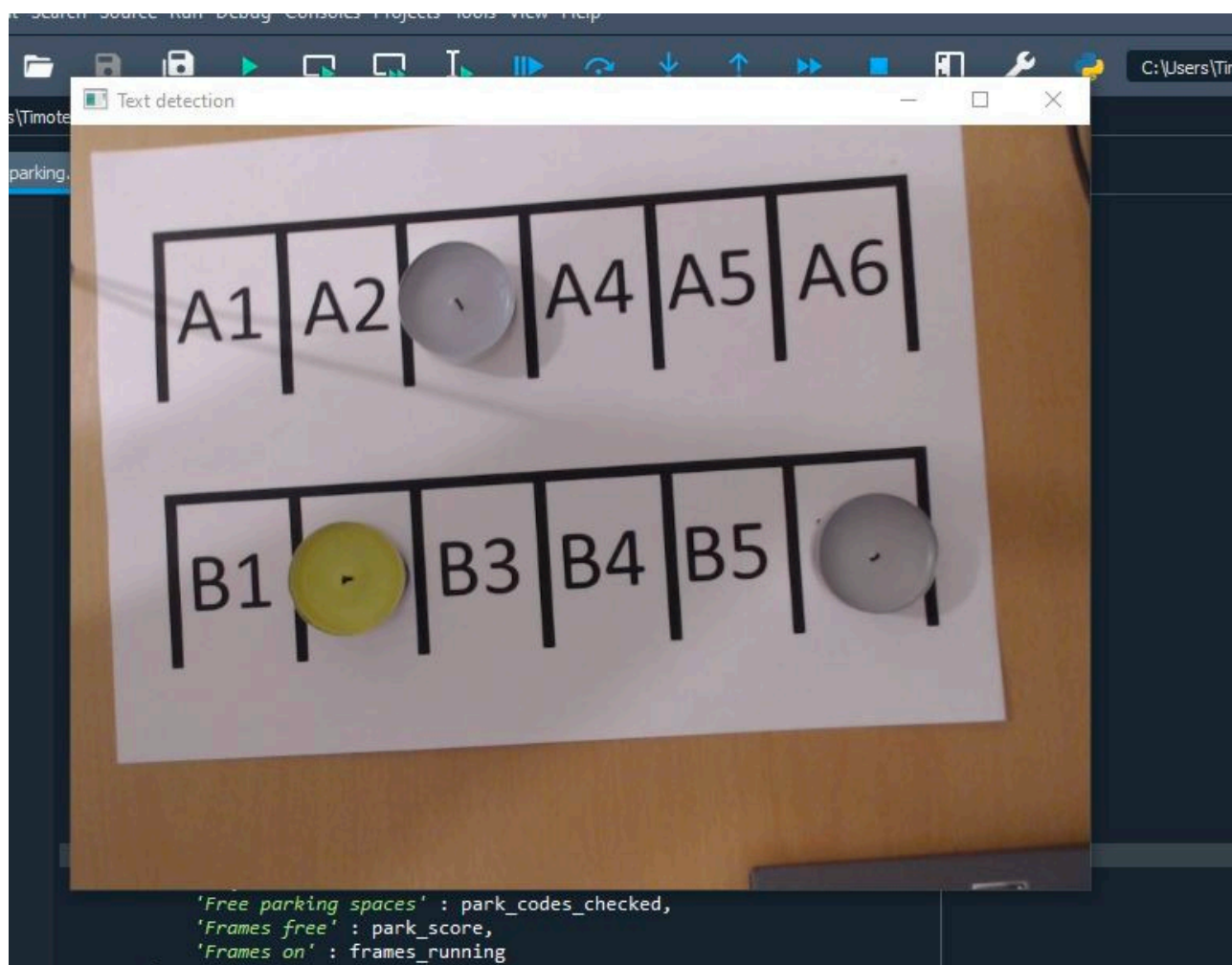
        # add free score to parking slot
        for code in park_codes_checked:
            if code not in park_score.keys():
                park_score[code] = 1
            else:
                park_score[code] = park_score[code] + 1

    # update firebase database
    data = {
        'Free parking spaces' : park_codes_checked,
        'Frames free' : park_score,
        'Frames on' : frames_running
    }

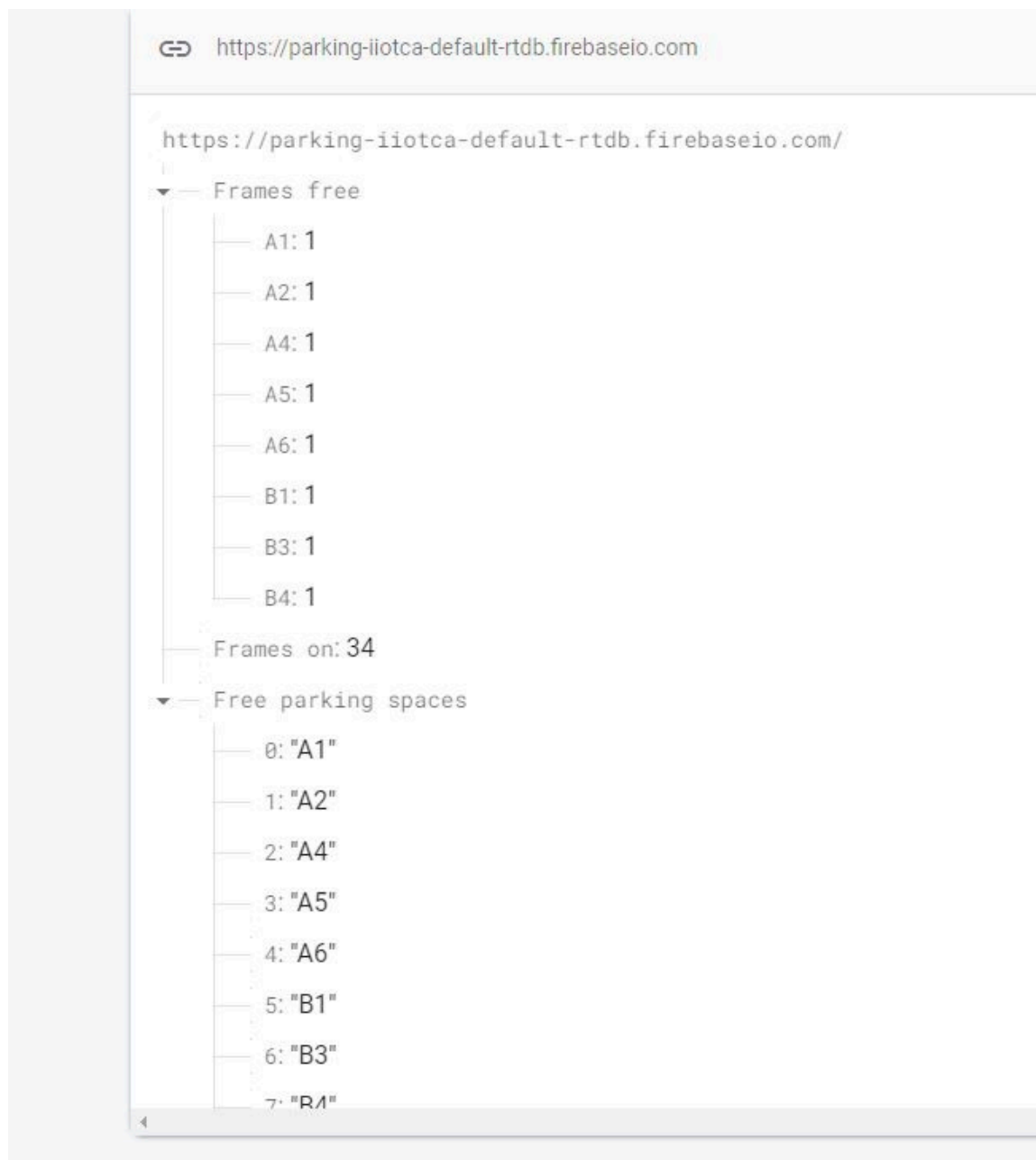
    db.update(data)
    print('Sent to Firebase')

# exit condition
if keyboard.is_pressed('a'):
    break
```

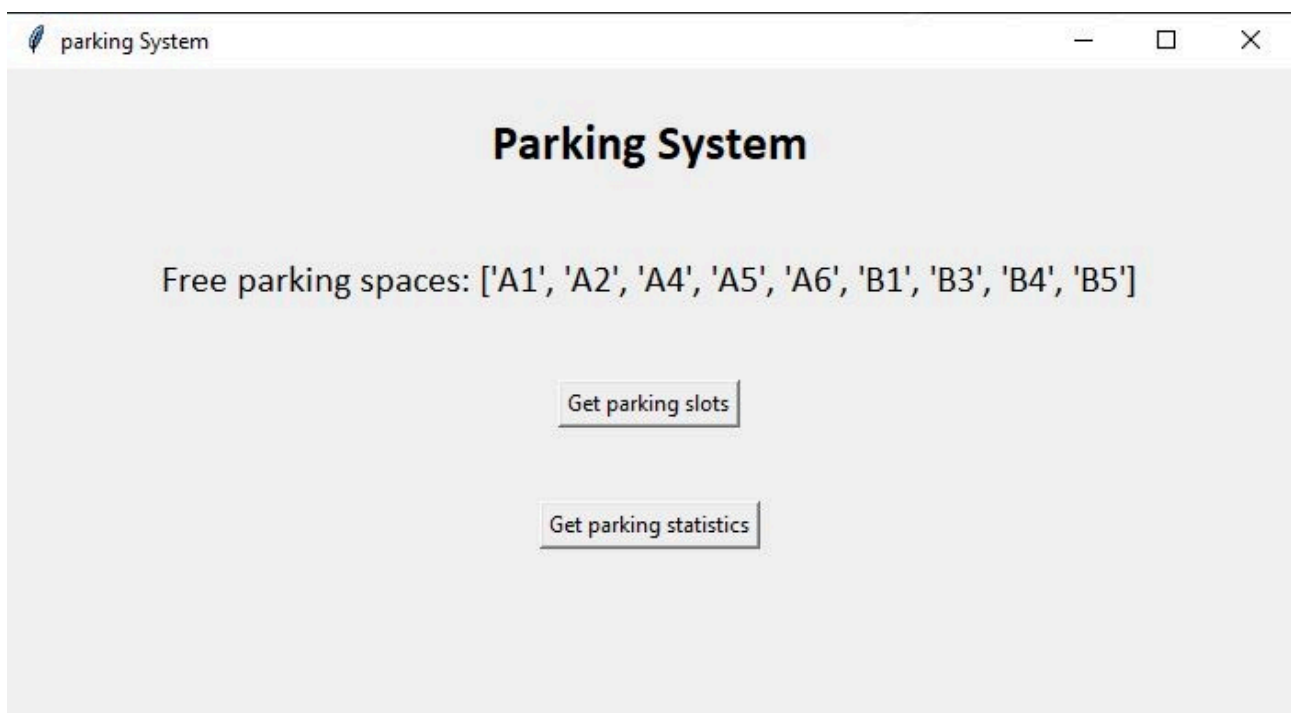
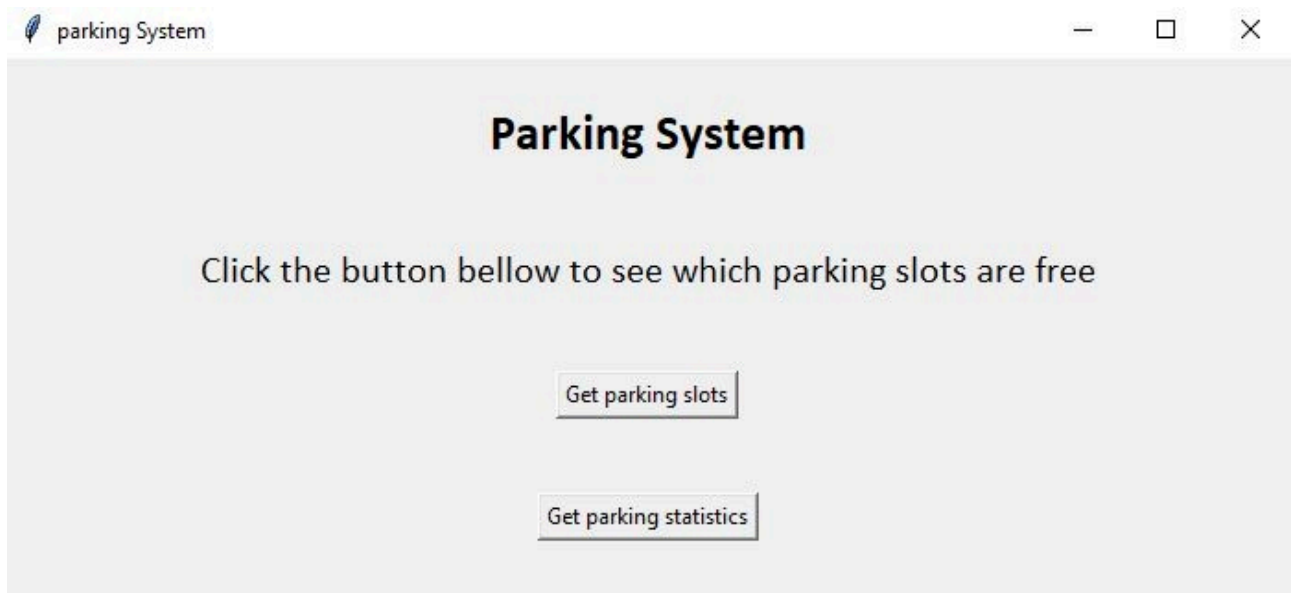
What is our camera seeing?



## Our Firebase database



## The app



## Technology Used

We wrote our program and compiled it in Spider by Anaconda using Python language.

To be able to use Python on Raspberry Pi[1] we installed and used OpenCV.

As the sensor we used a Logitech C920 1080p Web Camera.

We connect our database with Firebase Cloud[2] which we use as our “notification center”.

We also created an app[4] where you can see which parking slot is empty and also a statistic with every parking slot how many times it was empty.

## Future Work

In the future we can create a process where the AI can identify where is the car and which parking empty parking slot is closer.

## Our GitHub Repository

<https://github.com/Horiutu/ProiectIIOTCA>

## References

[1] <https://shorturl.at/ivwD2>

[2] <https://shorturl.at/joQV4>

[3] <https://shorturl.at/gFMOP>

[4] <https://shorturl.at/nvOV0>

[5] <https://shorturl.at/byAT3>

Thank You!