



"EASY PARK"

Tomas Shahwan 211397294

Yaniv Ridel 322568627

Noy Atedgi 211389887

Eti Perah 323838094



מבוא

הפרויקט עוסק בפיתוח מערכת חניה חכמה המבוססת על טכנולוגיות חיישני MQTT מרחק ותקשורת

המטרה

לזהות בזמן אמת האם חניה תפוסה או פנויה, ולהעביר את המידע בצורה אוטומטית לממשק משתמש נוח ויעיל

מה הבעיה שאנחנו פותרים?

חוסר מידע עדכני – הנהג לא יודע האם יש מקום עד שהוא מגיע.
חניונים ללא אוטומציה – אין ניטור אקטיבי או שמירה על נתונים.
מצוקת חניה גוברת בערים – חיפוש חניה מבזבז זמן, דלק ויוצר עומסים.
מיותרים מה שגורם לתסכול רב במציאת חניה





תיאור המערכת >>>>>>>>

מערכת EAZY PARK היא מערכת מבוססת אינטרנט של הדברים (IoT), אשר מאפשרת לזהות באופן אוטומטי את תפוסת מקום החניה ולהציג את המידע בצורה חכמה בזמן אמת.

המערכת כוללת חיישן מרחק הממוקם בכל מקום חניה ומודד את המרחק מכל עצם. כאשר רכב נכנס, החיישן מזהה שינוי ו"מבין" כי המקום תפוס.

המידע עובר דרך בקר (אמולטור) אשר מפרסם את הנתון לשרת MQTT בענן.

המידע מתקבל בממשק גרפי ייעודי (GUI) שבו ניתן לראות את סטטוס החניה: תפוס או פנוי.

בנוסף, כל ערך מרחק נשמר אוטומטית במסד נתונים מסוג SQLite, לצורכי תיעוד, מעקב וניתוחים עתידיים.

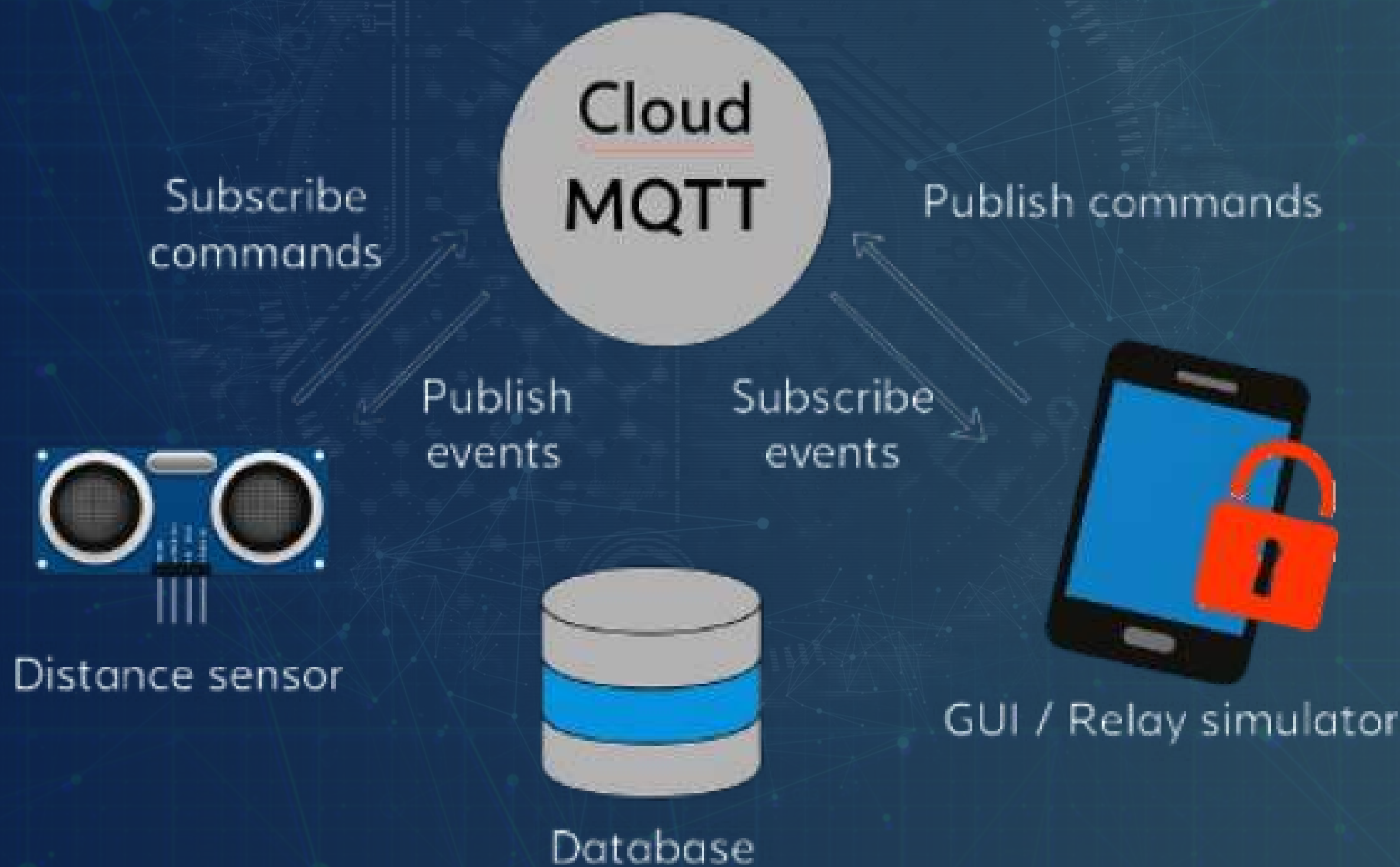
המערכת כוללת גם כפתור עם תאורת סטטוס (LED):

- כאשר החניה תפוסה – הכפתור נדלק באדום.
- כאשר החניה פנויה – הכפתור נדלק בירוק.
- כך ניתן לזהות בצורה מיידית את המצב גם ללא צורך במסך או אפליקציה.



עיצוב ותכנון

התרשים מייצג את מבנה המערכת הכולל של **Eazypark**, המבוססת על פרוטוקול **MQTT** בענן. חיישן המרחק שולח נתוני תפוס/פנוי לשרת **MQTT**, מהם ניזון ממשק המשתמש (**GUI**) שמציג את המידע בצורה ויזואלית כולל חיווי לד. כל הנתונים נרשמים במסד נתונים לצורכי תיעוד וניתוח עתידי.



שלבי פיתוח

פיתוח הנתונים
והפונקציונליות

```
56
57
58 def on_disconnect(self, client, userdata, flags, rc=0):
59     print("Disconnected result code "+str(rc))
60
61 def on_message(self, client, userdata, msg):
62     topic = msg.topic
63     m_decode = str(msg.payload.decode("utf-8", "ignore"))
64     print("message from: " + topic, m_decode)
65
66     try:
67         value = float(m_decode)
68         QObject.invokeMethod(mainwin, "save_distance", Qt.QueuedConnection, Q_ARG(float, value))
69     except ValueError:
70         pass
71
72     mainwin.subscribeDock.update_mess_win(m_decode)
73
74 def connect_to(self):
75     self.client = mqtt.Client(self.clientname, clean_session=True)
76     self.client.on_connect = self.on_connect
77     self.client.on_disconnect = self.on_disconnect
78     self.client.on_log = self.on_log
79     self.client.on_message = self.on_message
80     self.client.username_pw_set(self.username, self.password)
81     print("Connecting to broker ", self.broker)
82     self.client.connect(self.broker, self.port)
83
84 def disconnect_from(self):
85     self.client.disconnect()
86
87 def start_listening(self):
88     self.client.loop_start()
```

בדיקות יחידה



רכישת תוכנית לדדדד



בניית רכיבי מדידת מרחק
ונורות לחניה



הכלים לביצוע הפרויקט

1. MQTT פרוטוקול תקשורת להעברת הודעות קל משקל – שימש לשליחת מידע מהחיישן לממשק
2. PyQt5 בניית הממשק הגרפי
3. SQLite מסד נתונים פנימי – שמר את ערכי המרחק עם תאריך ושעה לבקרה עתידית
4. Random / Time סימולציה של מרחק משתנה של רכב – לדמות התקרבות ויציאה
5. Emulation לא השתמשנו בחיישן פיזי, אלא יצרנו אמולטור שמדמה את המרחק באופן אוטומטי
6. Event-Driven Logic כל שינוי במרחק הפעיל אירוע (DB-שמירה ל / GUI-שינוי ב / MQTT שליחת)



האמולטורים

1.

אמולטור מרחק

אמולטור המרחק מדמה מכשיר למדידת מרחק ברגע שמתקבל מרחק קטן מ 30 ס"מ מתעדכן שהחניה נתפסה בעזרת לחיצה על הכפתור שמחליף את צבעו לאדום והורדת השער של החניה

2.

אמולטור כפתור לד

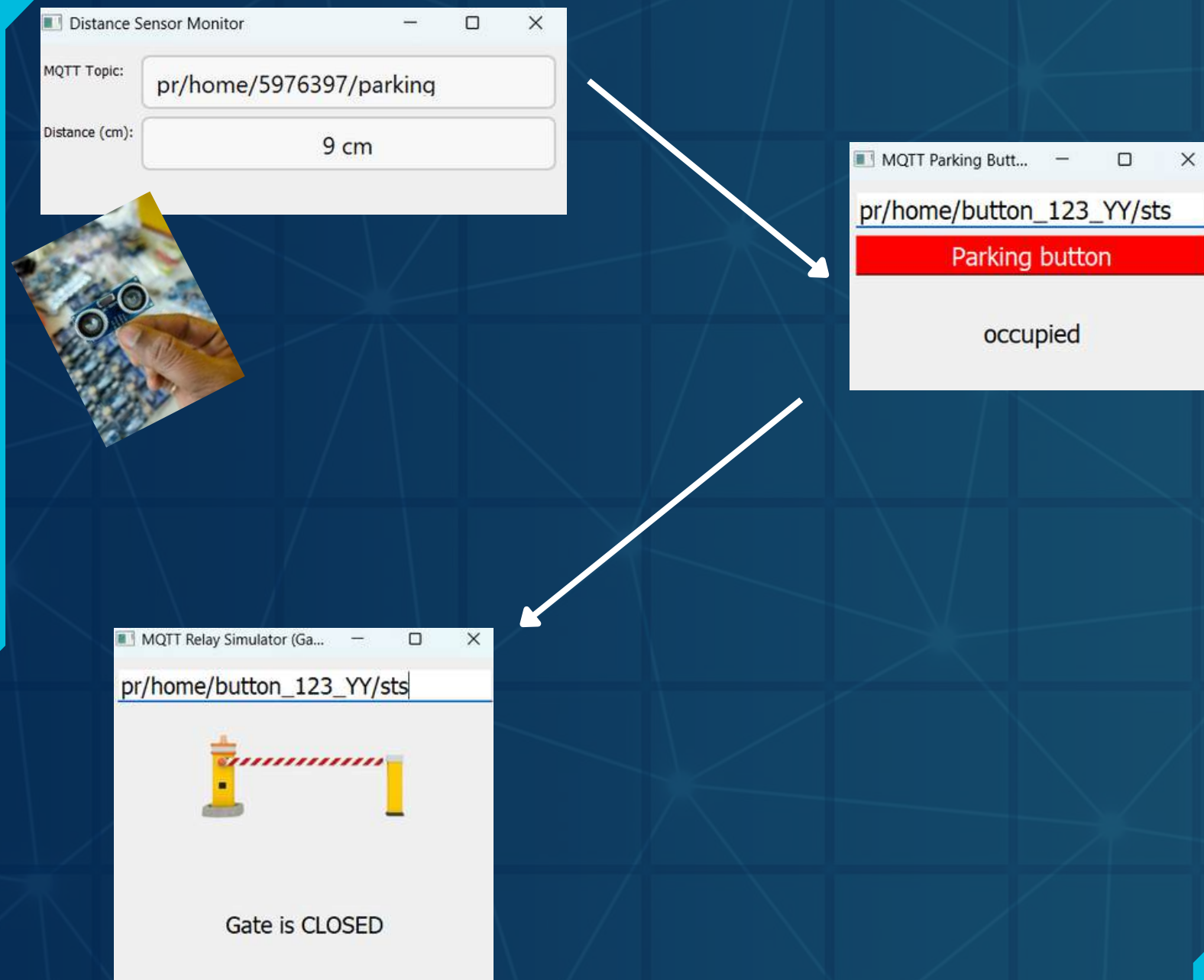
ברגע שבעזרת אמולטור המרחק מזוהה כי החניה נתפסה באופן אוטומטי אמולטור הכפתור מחליף את צבעו לאדום שמסמל שהחניה תפוסה ולאחר שמתפנה חוזר לירוק

3.

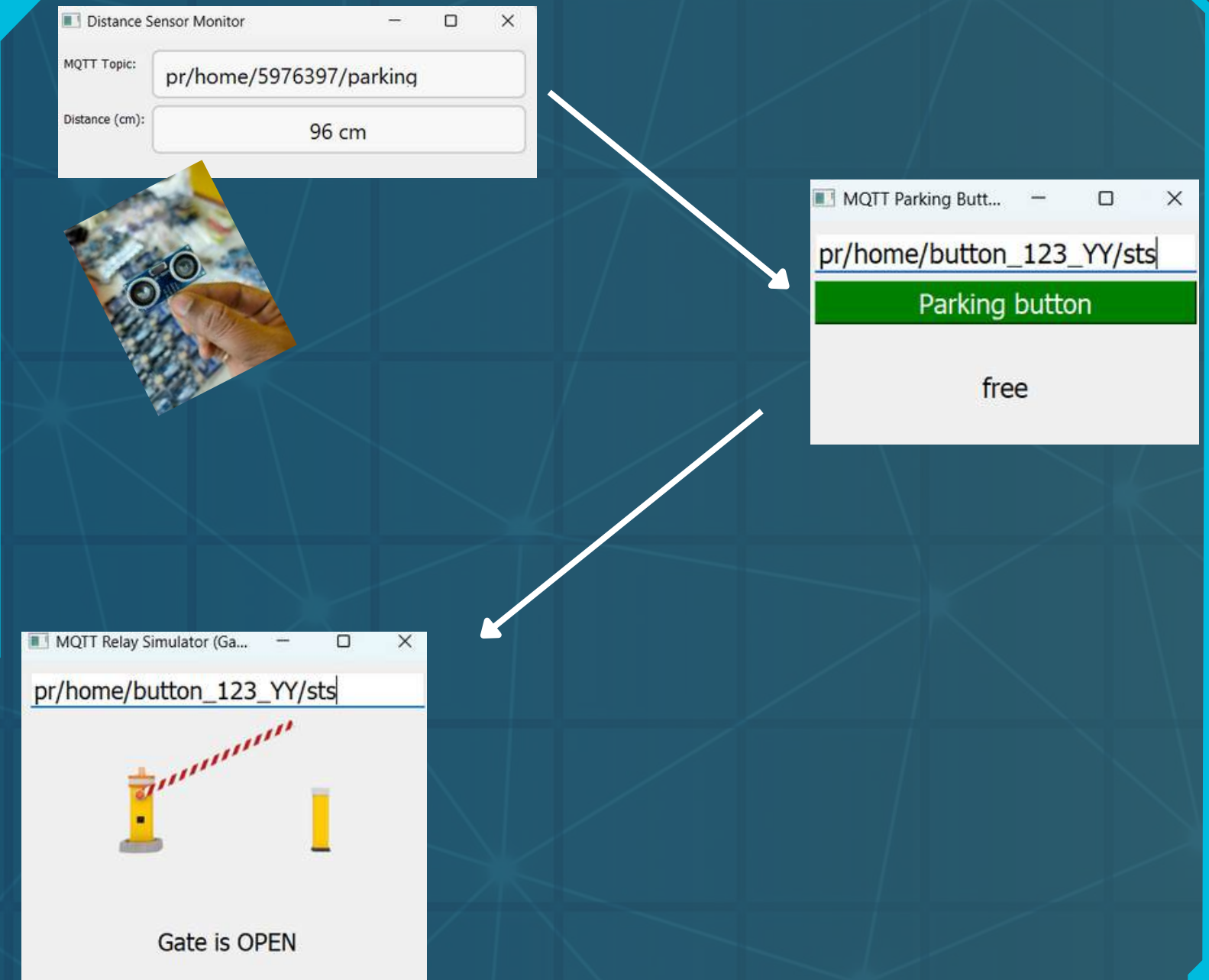
RELAY

ברגע שמזוהה שהחניה נתפסה תפקיד הריליי הוא לשלוח פקודה לסגור את השער ולפתוח ברגע שהחניה מתפנה

חניה נתפסת



חניה מתפנה



סיכום



בפרויקט זה פיתחנו מערכת חניה חכמה מבוססת IoT, שמטרתה לזהות ולדווח בזמן אמת על תפוסת מקום חניה.

המערכת כוללת

סימולציה של חיישן מרחק

תקשורת בענן באמצעות פרוטוקול MQTT

ממשק גרפי אינטואיטיבי שנבנה ב-PyQt

ושמירה של הנתונים במסד SQLite

באמצעות שימוש בטכנולוגיות פשוטות אך יעילות, הצלחנו להדגים פתרון אוטומטי, אמין ונגיש לבעיה יומיומית

המערכת תוכננה בצורה מודולרית, מה שמאפשר להתרחב בעתיד – לדוגמה, באמצעות שילוב אפליקציה לנייד או ניהול מספר חניות במקביל





MQTT פרוטוקול : mqtt

PyQT: GUI PyQt:

SQLite: מסד נתונים

קישור לפרויקט ב **GITHUB**

https://github.com/Tomasshahwan/IOT_FINAL_PROJECT

קישור לסרטון ב YOUTUBE

[HTTPS://YOUTU.BE/WVEMKGJ7G8M?](https://youtu.be/WVEMKGJ7G8M?)