



الجمهورية العربية السورية

معهد دمشق المتوسط

قسم المعلوماتية

دورة الاتصالات

نظام المرور النزي

مشروع تخرج أعدّ لنيل درجة الدبلوم في اختصاص
الاتصالات

إعداد الطلاب

رغد محمد

محمد محمد

روى الحارس

لجين شرورو

إشراف

م. وسيم الماضي

م. عبد الله سلامة

م. حذيفة محمد



اللهم فوضنا أمرنا إليك وألجأنا ظهركنا إليك ورغبة ورهبةً إليك،
لا ملجأ ولا منجا منك إلا إليك ... والصلاة والسلام على شفيعنا و
معلمنا رسول الله و الحمد لله رب العالمين.

كلمة شكر

بعد هذه السنين الدراسية نتقدّم بجزيل الشكر لمن بذلوا
الجهد ليقدّموا لنا من علمهم و تجاربهم وخبرتهم الطويلة .
و تكرموا بقبول الإشراف على مشروعنا و دعمنا حتى النهاية
م. حذيفة محمد م. عبد الله سلامة

ونخص بالشكر و الاحترام

المهندس وسيم الماضي

و ندعو الله أن يبارك في حياته و يجزيه عنا خير الجزاء .

فهرس المحتويات

8	المصطلحات
9	المقدمة
9	الهدف من المشروع
10	الفصل الأول : التعريف بالمشروع
11	المخطط الصندوقى لعمل المشروع
12	فكرة المشروع و آلية العمل
14	الفصل الثانى : الإطار النظري
18	الفصل الثالث : الإجراء العملي
24	المخطط التدفقي لخوارزمية المشروع
24	الأكواد البرمجية
28	الفصل الرابع : الاختبارات والنتائج
32	التنفيذ العملي للمشروع
32	آفاق مستقبلية
33	المراجع

المصطلحات

Abbreviation	Definition
OPENCV	open source computer vision
RTSP	Real-Time Streaming Protocol
IP Camera	Internet Protocol Camera
PSU	Power Supply Unit
AC	Alternating Current
DC	Direct Current
HTTP	Hyper Text Transfer Protocol
AI	Artificial Intelligence
YOLOv3	You Only Look Once version3
EEPROM	Electrically Erasable Programmable Read Only Memory
PWM	Pulse Width Modulation
USB	Universal Serial Bus
LAN	Local Area Network

المقدمة

في قلب المدن المزدحمة، تُشكل حركة المرور تحديًا يوميًا. مع تنامي عدد السكان وارتفاع معدل امتلاك السيارات، تتحول الطرق الرئيسية في المدن إلى شرايين مسدودة تعاني من الازدحام والاختناقات المرورية الشديدة.

ويُعزى جزء كبير من هذه الاختناقات إلى نظام إشارات المرور التقليدية الذي لم يعد بإمكانه إدارة وتنظيم حركة المرور بشكل سلس. لقد عفا الزمن على هذه الإشارات ولم تتطور مع تطور المدن. ومن هنا جاءت فكرة إشارة المرور الذكية، وهي حل مبتكر لتنظيم حركة المرور وتقليل الازدحام

الهدف من المشروع

تصميم نظام ذكي لإشارات المرور يستخدم معالجة الصور وتقنية آردوينو لتكييف أوقات الإشارات وفقًا لكثافة حركة المرور، بهدف

- تحسين انسيابية حركة السير وتقليل الاختناقات المرورية.
- تقليل الوقت الضائع في انتظار الإشارة الحمراء.
- تعزيز السلامة على الطرق.
- التقليل من انبعاثات المركبات.

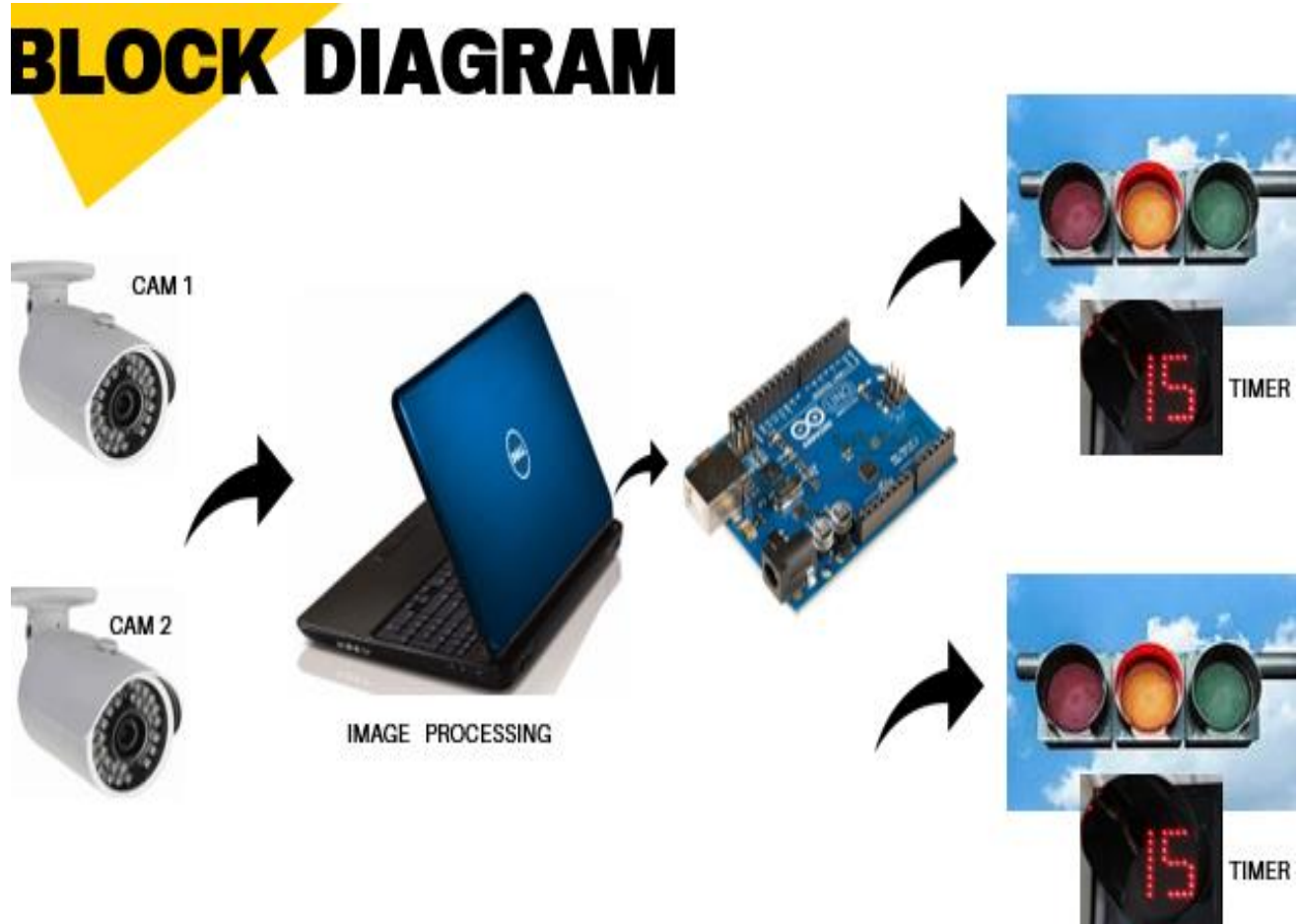
التأثير المتوقع

نسعى من خلال مشروعنا إلى وضع حجر الأساس لمدن ذكية تتميز بشوارع أكثر سلاسة وأمانًا.

الفصل الأول : التعريف بالمشروع

Chapter 1 : Introduction

المخطط الصندوقي لعمل المشروع:



الشكل 1-1 المخطط المبسط لعمل المشروع

فكرة المشروع:

نظام ذكي لإدارة حركة المرور يعتمد على الرؤية الحاسوبية و Arduino Mega .
يكتشف النظام ويحسب عدد السيارات في مسارين، ويحدد المسار الأكثر ازدحامًا،
و يمنحه الأولوية للمرور "الضوء الأخضر " لمدة تتناسب مع عدد السيارات.

آلية عمل المشروع :

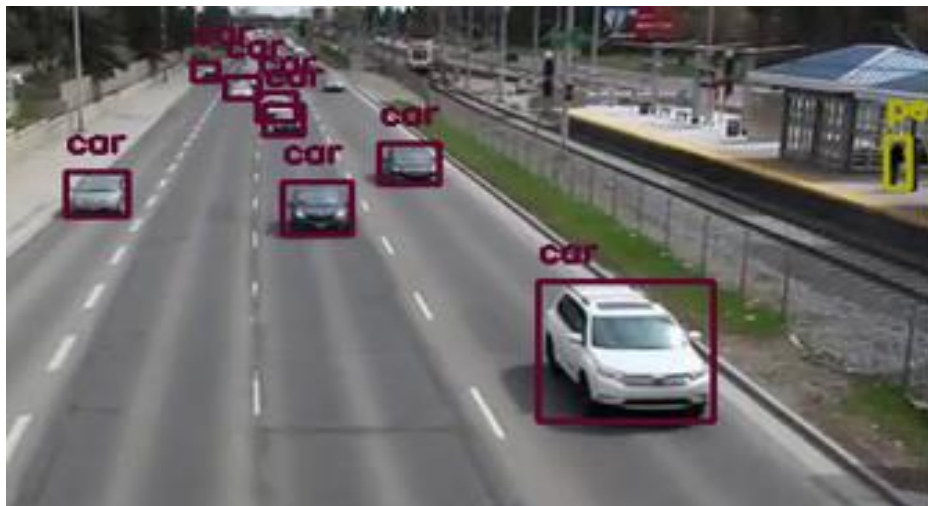
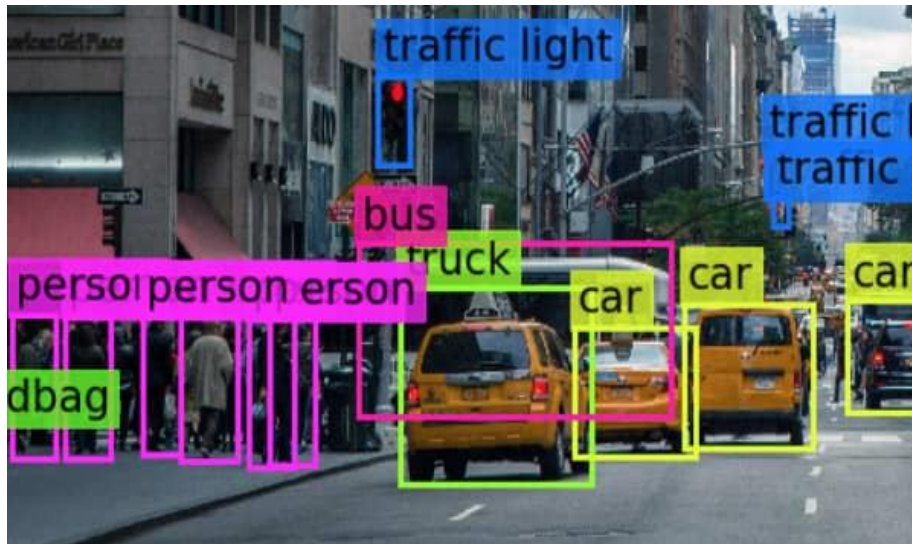
يستخدم النظام كاميرتين لتسجيل مقاطع فيديو لحركة المرور من مسارين مختلفين
باستخدام بروتوكول RTSP
يلتقط الكمبيوتر إطارات من تدفقات الفيديو للمسارين، ويستخدم OpenCV و
cvlib لاكتشاف السيارات وحساب عددها في كل مسار.
بعد ذلك يتم حساب كثافة حركة المرور في كل مسار من خلال عدد السيارات في
كل مسار.

يتم تحديد المسار الذي يحظى بأعلى كثافة (الأكثر ازدحامًا) ثم إعطاؤه الأولوية
للمرور "الضوء الأخضر".

يتم حساب مدة إشارة المرور الخضراء للمسار الأكثر ازدحامًا بناءً على عدد

السيارات "Real _time traffic light" و بحد أقصى 50 ثانية
يتحكم Arduino Mega في إضاءة مصابيح إشارات المرور الخضراء والصفراء
والحمراء لكل مسار وفقاً للمدة المحسوبة و تُعرض هذه المدة على شاشة

. 7-Segments



الفصل الثاني : الإطار النظري

Chapter 2 : Theoretical study

1-1 كاميرات IP السلكية

هي كاميرات رقمية ترسل البيانات عبر الشبكة المحلية و هي تختلف عن كاميرات المراقبة التناظرية التقليدية.

يستخدم مشروعا كاميرتي IP متصلتين بجهاز كمبيوتر مضيف عبر كابل إيثرنت يتطلب هذا الاتصال مصدر طاقة مستقرًا، عادةً 12 فولت تيار مستمر والذي يتم توفيره بواسطة جهاز power supply لتشغيل الكاميرات. توفر كاميرات IP جودة فيديو عالية، مما يسمح بتحديد هوية الأشخاص والأشياء بوضوح حيث يمكن دمج الكاميرات مع خوارزميات الرؤية الحاسوبية المتطورة .

يتم بث تغذية الفيديو من الكاميرات إلى جهاز الكمبيوتر المضيف باستخدام بروتوكول البث المباشر RTSP، مما يسمح بتسجيل اللقطات لكل مسار بالوقت الفعلي.

1-2 power supply unit (PSU)

وحدة الكترونية توفر الطاقة اللازمة لتشغيل مكونات الكاميرا، مثل المستشعر والعدسات والشاشة تحول التيار المتردد (AC) من مصدر الطاقة الخارجي إلى تيار مستمر (DC).

Router 1-3

يعمل الراوتر كجهاز تحويل، مما يتيح للكاميرات والكمبيوتر التواصل عبر شبكة محلية

1-4 بروتوكول RTSP

بروتوكول للتحكم في تدفق الوسائط عبر الشبكات. وهو يعمل فوق بروتوكول نقل النص التشعبي (HTTP) ؛ يستخدم بروتوكول RTSP على نطاق واسع في تطبيقات البث الحي ومراقبة الفيديو.

2-1 Computer vision

تستخدم أنظمة الرؤية الحاسوبية تقنية الذكاء الاصطناعي (AI) لتقليد قدرات الدماغ البشري في التعرف على الأشياء وتصنيفها. يقوم علماء الكمبيوتر بتدريب أجهزة الكمبيوتر على فهم البيانات المرئية عن طريق إدخال كميات هائلة من المعلومات. تحدد خوارزميات تعلم الآلة الأنماط الشائعة في هذه الصور أو مقاطع الفيديو وتطبق هذه المعرفة لتحديد الصور غير المعروفة بدقة. على سبيل المثال، إذا تمت معالجة ملايين صور السيارات بواسطة أجهزة الكمبيوتر، فإنها ستبدأ في بناء أنماط هوية يمكنها اكتشاف سيارة بدقة في صورة.

2-2 نموذج YOLOv3

نموذج شبكة عصبية للكشف عن الأشياء في الصور والفيديوهات في الوقت الفعلي بدقة عالية وسرعة. تم تطويره بواسطة جوزيف ريدمون وزملائه في عام 2018.

يستخدم نموذج YOLOv3 نهجاً فريداً للكشف عن الأشياء يعرف باسم "التنبؤات على شبكة أحادية الطلقة". حيث تقوم الشبكة بتقسيم الصورة إلى خلايا وتتوقع ما إذا كانت كل خلية تحتوي على كائن وما هو الإطار المحيط بهذا الكائن.

3-1 Arduino Mega

لوحة تطوير إلكترونية قائمة على متحكم دقيق Atmel ATmega2560 . مصممة للمشاريع التي تتطلب عددًا كبيرًا من المدخلات والمخرجات الرقمية والتناظرية. ميزاتها :

- متحكم دقيق Atmega2560 مع 256 كيلو بايت من ذاكرة الفلاش و 8 كيلو بايت من ذاكرة الوصول العشوائي و 4 كيلو بايت من ذاكرة EEPROM
- 54 دبوس إدخال/إخراج رقمي (منها 15 دبوس PWM)
- 16 دبوس إدخال/إخراج تناظري

في مشروعنا يتحكم Arduino Mega في إضاءة مصابيح إشارات المرور الخضراء والصفراء والحمراء لكل مسار وفقًا للمدة المحسوبة و تُعرض هذه المدة على شاشتي 7-Segments

3-2 PyFirmata2

مكتبة برمجيات مفتوحة المصدر مبنية على بايثون تسمح بالتواصل مع لوحات Arduino من خلال منفذ تسلسلي منفذ USB

3-2 7-Segments

شاشة عرض تتكون من 7 أجزاء أو وحدات عرض فردية، كل منها يشكل جزءًا من رقم أو حرف.

الفصل الثالث : الإجراء العملي

Chapter 3 : Practical implementation

خوارزمية التنفيذ

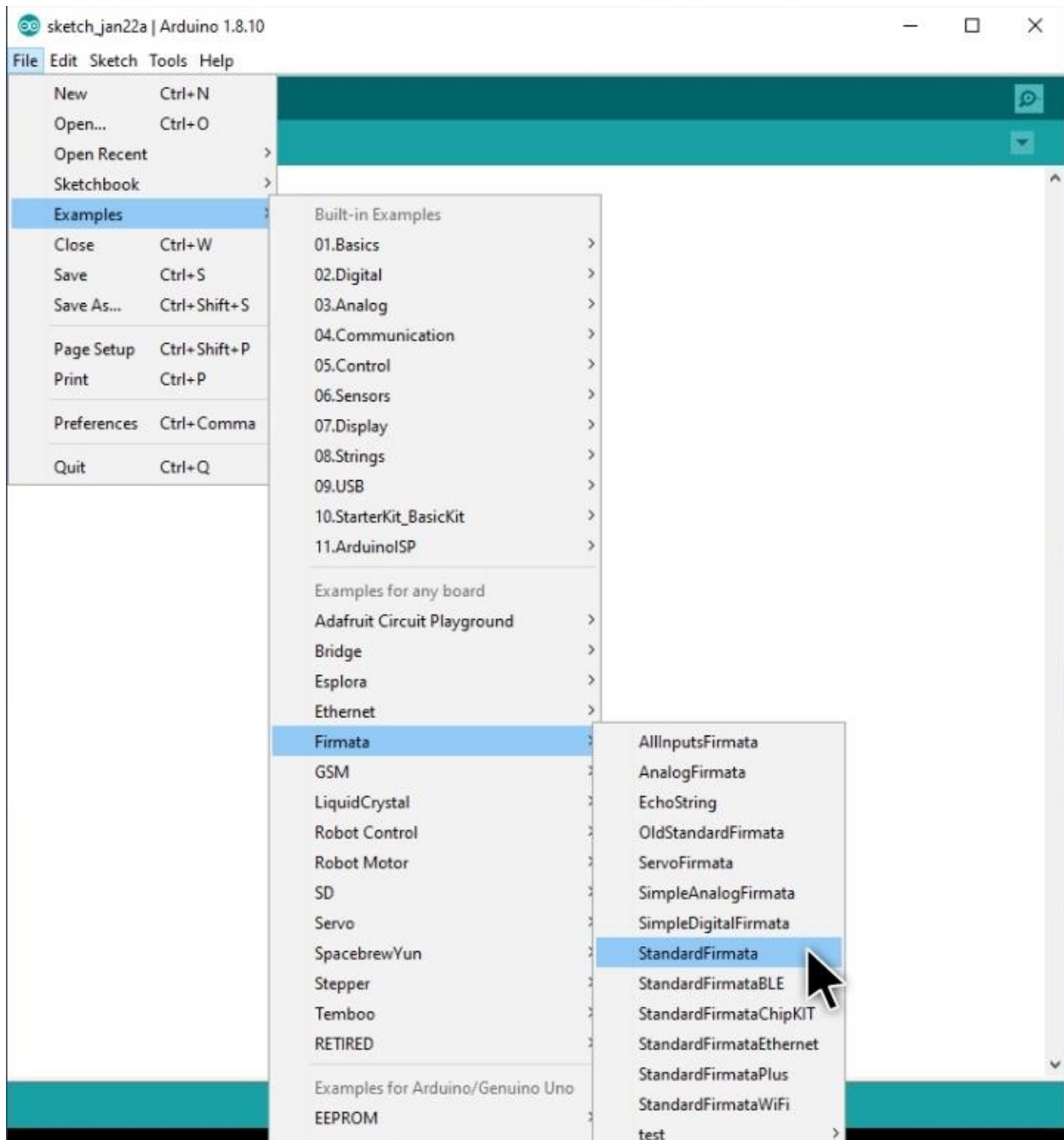
توصيل كاميرات IP

1. قمنا بتوصيل كابل الطاقة 12 فولت بكاميرتي IP.
2. قمنا بتوصيل كابلات شبكة إيثرنت بكاميراتي IP.
3. قمنا بتوصيل كابل شبكة إيثرنت من منفذ LAN بالراوتر إلى منفذ إيثرنت في الكمبيوتر.

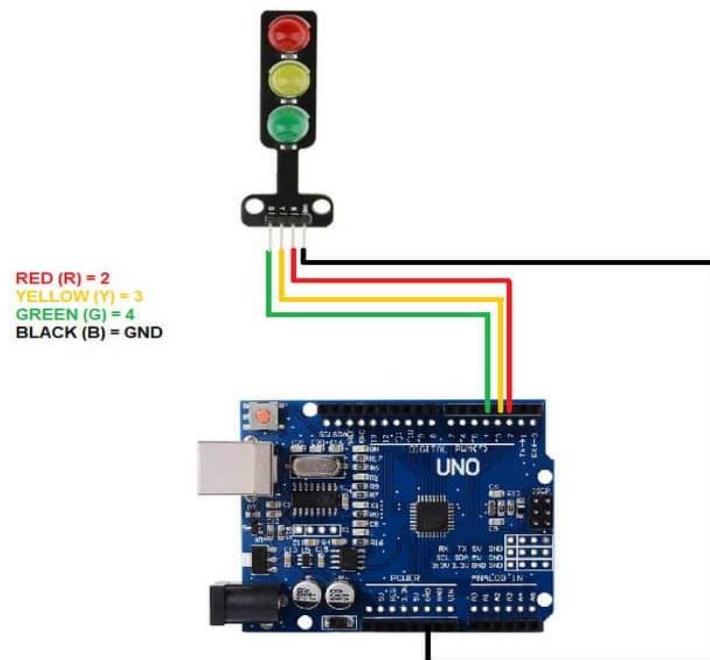
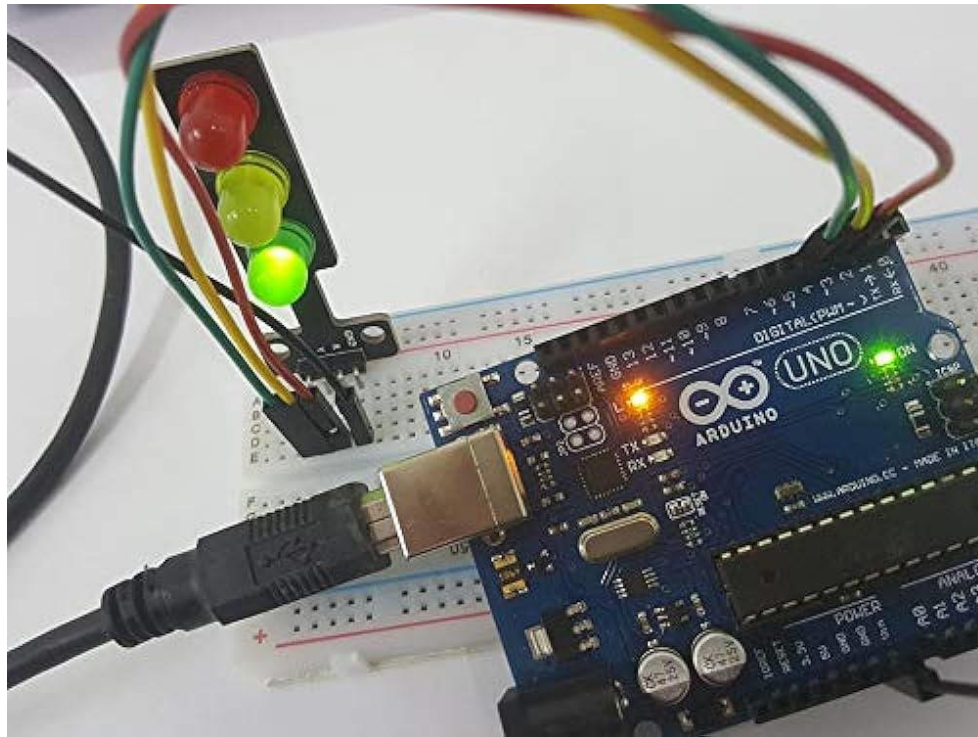


تهيئة الأردوينو :

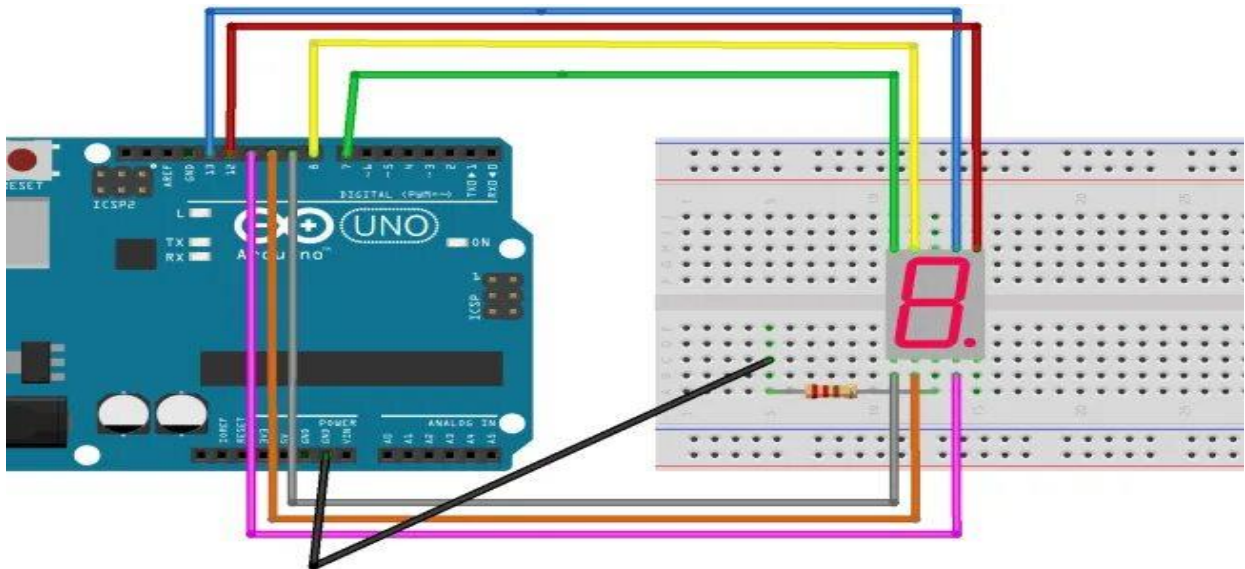
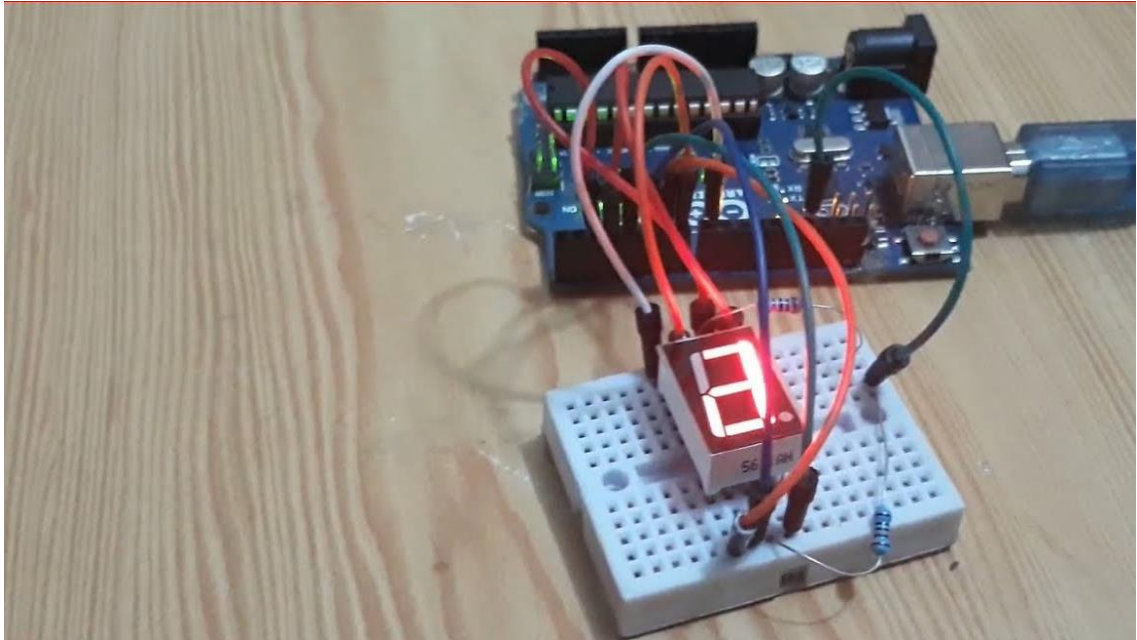
نقوم بتحميل مكتبة standard Firmata على الأردوينو أولاً .



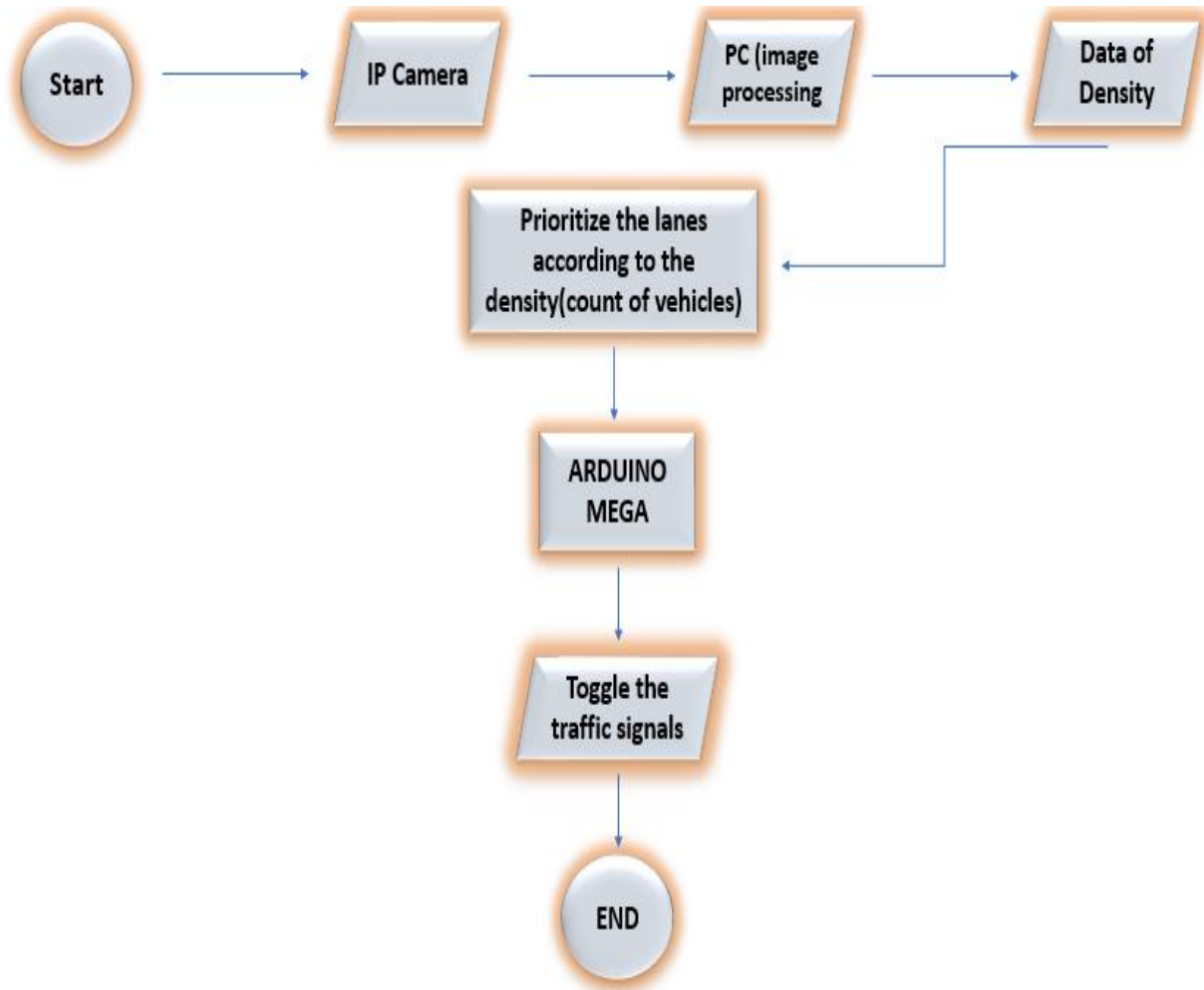
توصيل موديول اشارة المرور إلى الأردوينو



توصيل شاشة 7-Segments إلى الأردوينو



المخطط التدفقي لخوارزمية المشروع :



الأكواد البرمجية

```
1 import time
2 import pyfirmata2
3 import time
4 import cv2
5 import matplotlib.pyplot as plt
6 import cvlib as cv
7 from cvlib.object_detection import draw_bbox
8 board = pyfirmata2.ArduinoMega('COM9')
9 while(True):
10
11     # Video 1
12     # Open the video file
13     rtsp_link1 = 'rtsp://admin:admin12345@192.168.1.108:554/live'
14     cap = cv2.VideoCapture(rtsp_link1)
15
16     # Check if video capture is successful
17     if not cap.isOpened():
18         print("Error opening video file")
19         exit()
20
21     # Define the number of seconds to skip
22     skip_seconds = 1
23
24     # Skip the specified number of seconds
25     cap.set(cv2.CAP_PROP_POS_MSEC, skip_seconds * 1000)
26
27     # Read the frame
28     ret, frame = cap.read()
29
30     # If frame is empty, video capture is over
31     if not ret:
32         print("Error reading frame from video file")
33         exit()
34
35     # Detect objects in the frame
36     bbox, label, conf = cv.detect_common_objects(frame)
37
38     # Count the number of cars detected
39     car_count1 = label.count('car')
40
41     # Draw bounding boxes around the detected objects
42     output_image = draw_bbox(frame, bbox, label, conf)
43
44     # Display the frame with bounding boxes around cars
45     cv2.imshow('Cars Detected in Lane 1', output_image)
46     cv2.waitKey(0)
47
48     # Store the frame in a variable
49     im = frame
50
51     # Print the number of cars detected in lane 1
52     print('Number of vehicles in lane 1 in the image is ', car_count1)
```



```
1  # Video 2
2  # Open the video file
3  rtsp_link2= 'rtsp://admin:admin12345@192.168.1.109:554/live'
4  cap = cv2.VideoCapture(rtsp_link2)
5  # Check if video capture is successful
6  if not cap.isOpened():
7      print("Error opening video file")
8      exit()
9  # Define the number of seconds to skip
10 skip_seconds = 7
11 # Skip the specified number of seconds
12 cap.set(cv2.CAP_PROP_POS_MSEC, skip_seconds * 1000)
13 # Read the frame
14 ret, frame = cap.read()
15 # If frame is empty, video capture is over
16 if not ret:
17     print("Error reading frame from video file")
18     exit()
19 # Detect objects in the frame
20 bbox, label, conf = cv.detect_common_objects(frame)
21 # Count the number of cars detected
22 car_count2 = label.count('car')
23 # Draw bounding boxes around the detected objects
24 output_image = draw_bbox(frame, bbox, label, conf)
25 # Display the frame with bounding boxes around cars
26 cv2.imshow('Cars Detected in Lane 2', output_image)
27 cv2.waitKey(0)
28 # Store the frame in a variable
29 im = frame
30 # Print the number of cars detected in lane 2
31 print('Number of vehicles in lane 2 in the image is ', car_count2)
```

```

1  def DURATION():
2      sorted_density = Find_density()
3      max_density_lane = sorted_density[0]
4      if max_density_lane[1] == 0:
5          print(f'{max_density_lane[0]} turns green for 0 sec!')
6          return 0, max_density_lane[0]
7      else:
8          # احسب وقت التشغيل بالثواني .
9          duration = min(round(max_density_lane[1] * 2), 50)
10         print(f'{max_density_lane[0]} turns green for {duration} sec!')
11         return duration, max_density_lane[0]
12     # استدعاء الدالتين
13     sorted_density = Find_density()
14     duration, max_density_lane = DURATION()
15     SEG_PINS2 = [30, 31, 32, 33, 34, 35, 36] # الأولى Segment-منافذ شاشة 7
16     COMMON_PIN2 = 5
17     SEG_PINS = [ 6, 7, 8, 9, 10, 11, 12 ] # 7 Segment-منافذ شاشة
18     COMMON_PIN = 13 # افتراض أنه 13، قم بتغييره إذا لزم الأمر)
19     # تعريف أنماط الأرقام
20     # المضاءة لرقم معين segments كل قائمة تمثل (a, b, c, d, e, f, g)
21     NUMBERS2 = {
22     0: [1, 1, 1, 1, 1, 1, 0],
23     1: [0, 1, 1, 0, 0, 0, 0],
24     2: [1, 1, 0, 1, 1, 0, 1],
25     3: [1, 1, 1, 1, 0, 0, 1],
26     4: [0, 1, 1, 0, 0, 1, 1],
27     5: [1, 0, 1, 1, 0, 1, 1],
28     6: [1, 0, 1, 1, 1, 1, 1],
29     7: [1, 1, 1, 0, 0, 0, 0],
30     8: [1, 1, 1, 1, 1, 1, 1],
31     9: [1, 1, 1, 1, 0, 1, 1],
32     }
33     NUMBERS = {
34     0: [1, 1, 1, 1, 1, 1, 0],
35     1: [0, 1, 1, 0, 0, 0, 0],
36     2: [1, 1, 0, 1, 1, 0, 1],
37     3: [1, 1, 1, 1, 0, 0, 1],
38     4: [0, 1, 1, 0, 0, 1, 1],
39     5: [1, 0, 1, 1, 0, 1, 1],
40     6: [1, 0, 1, 1, 1, 1, 1],
41     7: [1, 1, 1, 0, 0, 0, 0],
42     8: [1, 1, 1, 1, 1, 1, 1],
43     9: [1, 1, 1, 1, 0, 1, 1],
44     }
45     # OUTPUT تعريف المنافذ كـ
46     for pin in SEG_PINS2 + [COMMON_PIN2]:
47         board.digital[pin].mode = pyfirmata2.OUTPUT
48     for pin in SEG_PINS + [COMMON_PIN]:
49         board.digital[pin].mode = pyfirmata2.OUTPUT
50     # دالة لعرض رقم على الشاشة الأولى
51     def display_number1(number):
52         segments = NUMBERS.get(number, [0, 0, 0, 0, 0, 0, 0])
53         for i, pin in enumerate(SEG_PINS):
54             board.digital[pin].write(segments[i])
55             board.digital[COMMON_PIN].write(0)
56     # دالة لعرض رقم على الشاشة الثانية
57     def display_number2(number):
58         segments = NUMBERS2.get(number, [0, 0, 0, 0, 0, 0, 0])
59         for i, pin in enumerate(SEG_PINS2):
60             board.digital[pin].write(segments[i])
61         board.digital[COMMON_PIN2].write(0)

```

```

1  def Ard_function():
2
3      time1 = 1 # 3 ثوانٍ time1 حددت متغير
4      pins = {
5          'lane-2': {'green': 22, 'yellow': 23, 'red': 24},
6          'lane-1': {'green': 25, 'yellow': 26, 'red': 27}
7      }
8      def control_lights():
9          for pin in [23, 26]: # تشغيل مصابيح
10             board.digital[pin].write(1)
11             time.sleep(time1) # إبقاء LED لمدّة 1 ثوانٍ
12             for pin in [23, 26]: # إطفاء مصابيح
13                 board.digital[pin].write(0)
14
15             # الأخضر للمسار الأكثر كثافة LED تشغيل مصباح
16             board.digital[pins[max_density_lane]['green']].write(1)
17             # الأحمر للمسار الآخر LED تشغيل مصباح
18             other_lane = 'lane-1' if max_density_lane == 'lane-2' else 'lane-2'
19             board.digital[pins[other_lane]['red']].write(1)
20             def countdown(start_number):
21                 for i in range(start_number, -1, -1):
22                     display_number1(i)
23                     display_number2(i)
24                     time.sleep(1)
25             # مع رقم البداية main_loop استدعاء الدالة
26             # قم بتغيير هذا الرقم حسب الرغبة
27             main_loop(duration) # قم بتغيير هذا الرقم حسب الرغبة
28             # إطفاء جميع مصابيح LED
29             for pin in [22,23,24,25,26,27]: # إطفاء مصابيح
30                 board.digital[pin].write(0)
31             control_lights() # استدعاء الدالة max_density_lane في مصابيح
32             Ard_function()

```

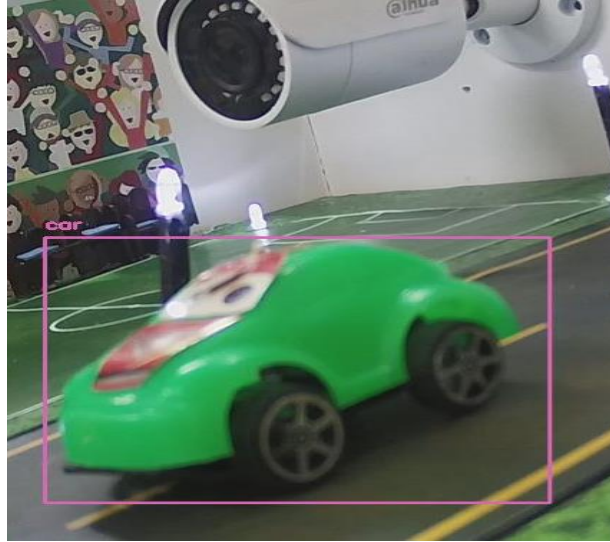

الفصل الرابع : الإختبارات و النتائج

Chapter 4 : Tests and results

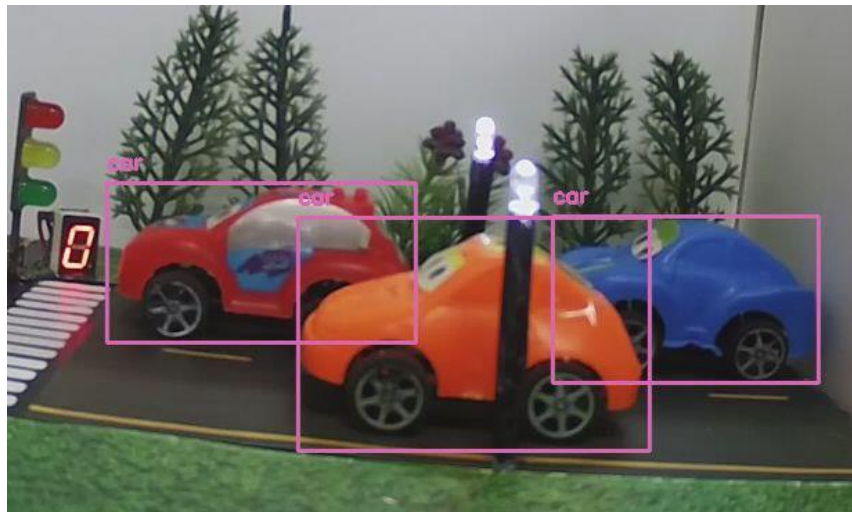
التنفيذ العملي للمشروع :



تم تطبيق النظام على مجسم يحتوي على مسارين
 إذا كان عدد السيارات في المسار الأول (1)
 وعدد السيارات في المسار الثاني (3) كما هو موضح في الصور التالية:



الشكل (1) المسار الأول



الشكل (2) المسار الثاني

فإنه يتم تخصيص وقت الإشارة الخضراء للمسار الثاني قدره (6) ثواني بمعدل ثانيتين لكل سيارة ، أما المسار الأول ستكون إشارته حمراء لوجود مركبة واحدة فقط.



الشكل (3) النتيجة النهائية

النتائج التجريبية

النتائج التجريبية التي تم الحصول عليها تظهر أن النظام قد خفض بشكل كبير وقت الانتظار عند الإشارة الحمراء؛ وذلك لأن النظام يتكيف وفقاً لكثافة حركة المرور .

في نظام إشارة المرور التقليدية نحتاج إلى الانتظار لمدة 30 ثانية على الأقل حتى إذا لم يكن هناك حركة مرور في جوانب أخرى من التقاطع . أما في إشارة المرور الذكية التي تم تطبيقها لم نعد نحتاج للوقوف على الإشارة الحمراء إذا لم يكن هناك حركة مرور في جوانب أخرى من التقاطع.

آفاق مستقبلية

بناء على نتائج نظام إشارة المرور الذكية نوصي ببعض الإضافات التي تساهم في تطوير هذا النظام .

- تطوير النظام لجعله يتغذى بالطاقة الشمسية .
- تطوير النظام ليشمل تقاطع أربع طرق وليس طريقين فقط .
- التقاط أرقام لوحات السيارات المخالفة لإشارة المرور عن طريق كاميرا .

المراجع

- 1] Prof.R.U.Yawle, Kiran.K.Modak, Parmeshwar.S.Shivshette, Snehal.S.Vhaval - Smart Traffic Control System - (SSRG-IJECE) - 03/03/2016.
- [2] A. ALBAGUL, H. HAMED, M. NAJI, A. ASSENI, A. ZARAGOUN
Design and Fabrication of a Smart Traffic Light Control System - Faculty of Electronic Technology, Baniwalid, LIBYA 2012.
- [3] Mohammed Ehsan Safi - Smart Traffic light controller based on Microcontroller - IJCCCE - 16/01/2016.
- [4] Bilal Ghazal, KhaledElkhatib, KhaledChahine, MohamadKherfan - Smart Traffic Light Control System - IEEE - 19/05/2016.
- [5] GurudattaVerma, RishabhSonkar, LekhrajBowaria - Smart Traffic Light System - IJSTE - April /2018.

Smart Traffic System

Graduation Project prepared to obtain Diploma in
Telecommunication

By

Mohammed Mhmd

Raghad Mhmd

Lujain Sharoro

Riwa Alharis

Supervised by

Eng. Abdullah Salama

Eng. Huthaifa Mhmd

Academic Year

2023 - 2024