ENME 809T

UMCP, Mitchell

- QR Codes

System Requirements

Using Raspberry Pi + camera:

1. Create QR Code & save to file
2. Create method to determine if QR Code is present in camera field of view
3. If detection:
   a) Bounding box
   b) Print QR Code to screen & terminal window

OpenCV

- QR Code functionality requires OpenCV 4

pi@raspberrypi: ~

```
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.4.4'
>>> exit()
pi@raspberrypi:~ $
```

OpenCV

- QR Code functionality requires OpenCV 4

pi@raspberrypi: ~

```
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.4.4'
>>> exit()
pi@raspberrypi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian stretch InRelease [25.4 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7
 MB]
Get:4 http://archive.raspberrypi.org/debian stretch/main armhf Packages [220 kB]
Get:5 http://archive.raspberrypi.org/debian stretch/ui armhf Packages [45.0 kB]
Get:6 http://raspbian.raspberrypi.org/raspbian stretch/contrib armhf Packages [5
6.9 kB]
Fetched 12.0 MB in 1min 2s (194 kB/s)
Reading package lists... Done
pi@raspberrypi:~ $
```

OpenCV

- QR Code functionality requires OpenCV 4

pi@raspberrypi: ~

```
pi@raspberrypi:~ $ sudo apt-get install libqt4-test python3-sip python3-pyqt5 li
bqtgui4 libjasper-dev libatlas-base-dev -y
```

OpenCV

- QR Code functionality requires OpenCV 4

pi@raspberrypi: ~

```
pi@raspberrypi:~ $ sudo pip3 install opencv-contrib-python==4.1.0.25
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting opencv-contrib-python==4.1.0.25
  Downloading https://www.piwheels.org/simple/opencv-contrib-python/opencv_contr
ib_python-4.1.0.25-cp35-cp35m-linux_armv7l.whl (15.8MB)
     |[redacted]| 15.9MB 126kB/s
Requirement already satisfied: numpy>=1.12.1 in /usr/lib/python3/dist-packages (
from opencv-contrib-python==4.1.0.25) (1.12.1)
Installing collected packages: opencv-contrib-python
Successfully installed opencv-contrib-python-4.1.0.25
WARNING: You are using pip version 19.1.1, however version 20.0.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
pi@raspberrypi:~ $
```

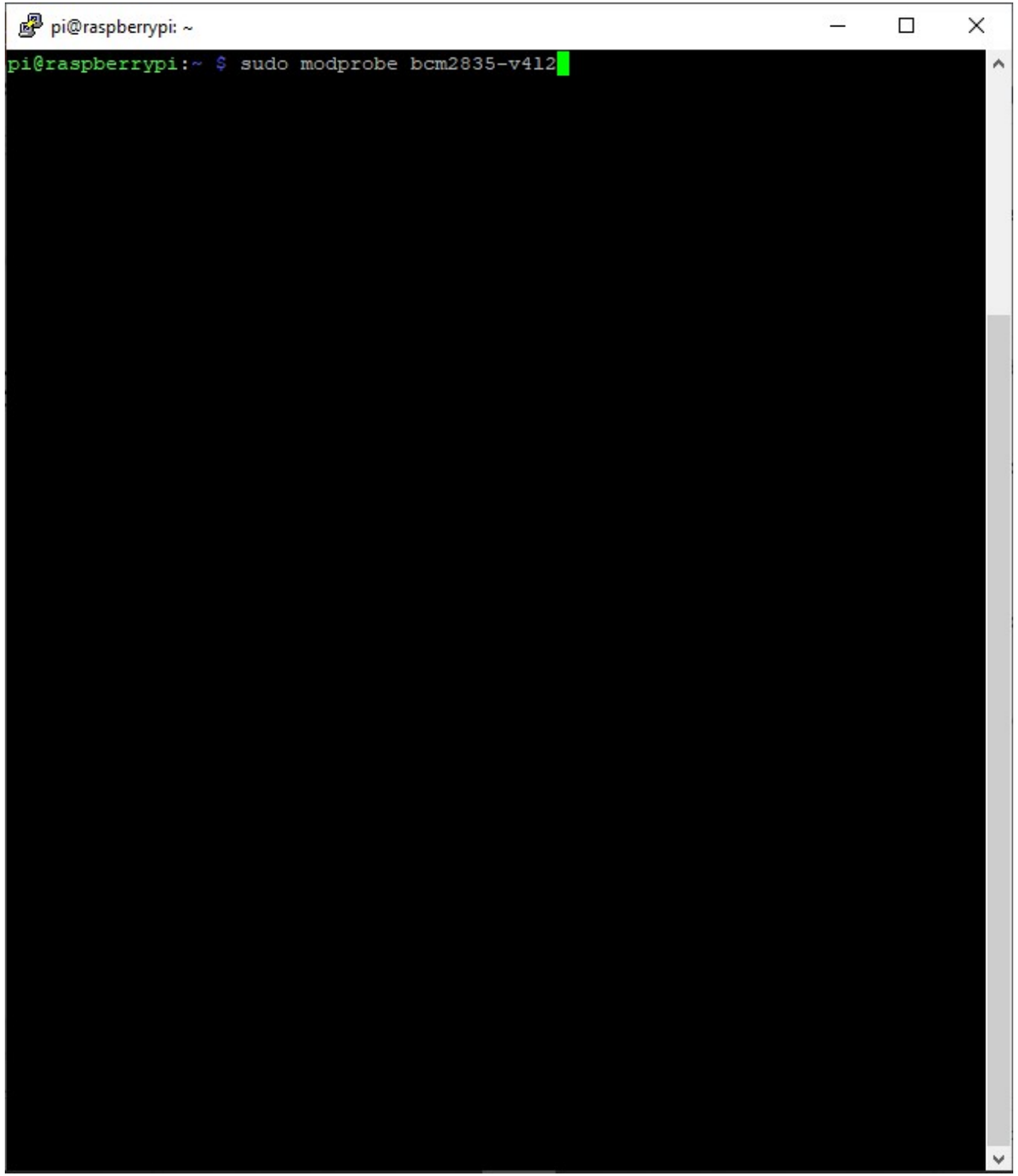
OpenCV

- QR Code functionality requires OpenCV 4

Note: this call is -4vl2

"dash 4 vee l-as-in-lima 2"

Permits use of the Pi camera with OpenCV VideoCapture



QR Codes

- Install qrcode package

pi@raspberrypi: ~

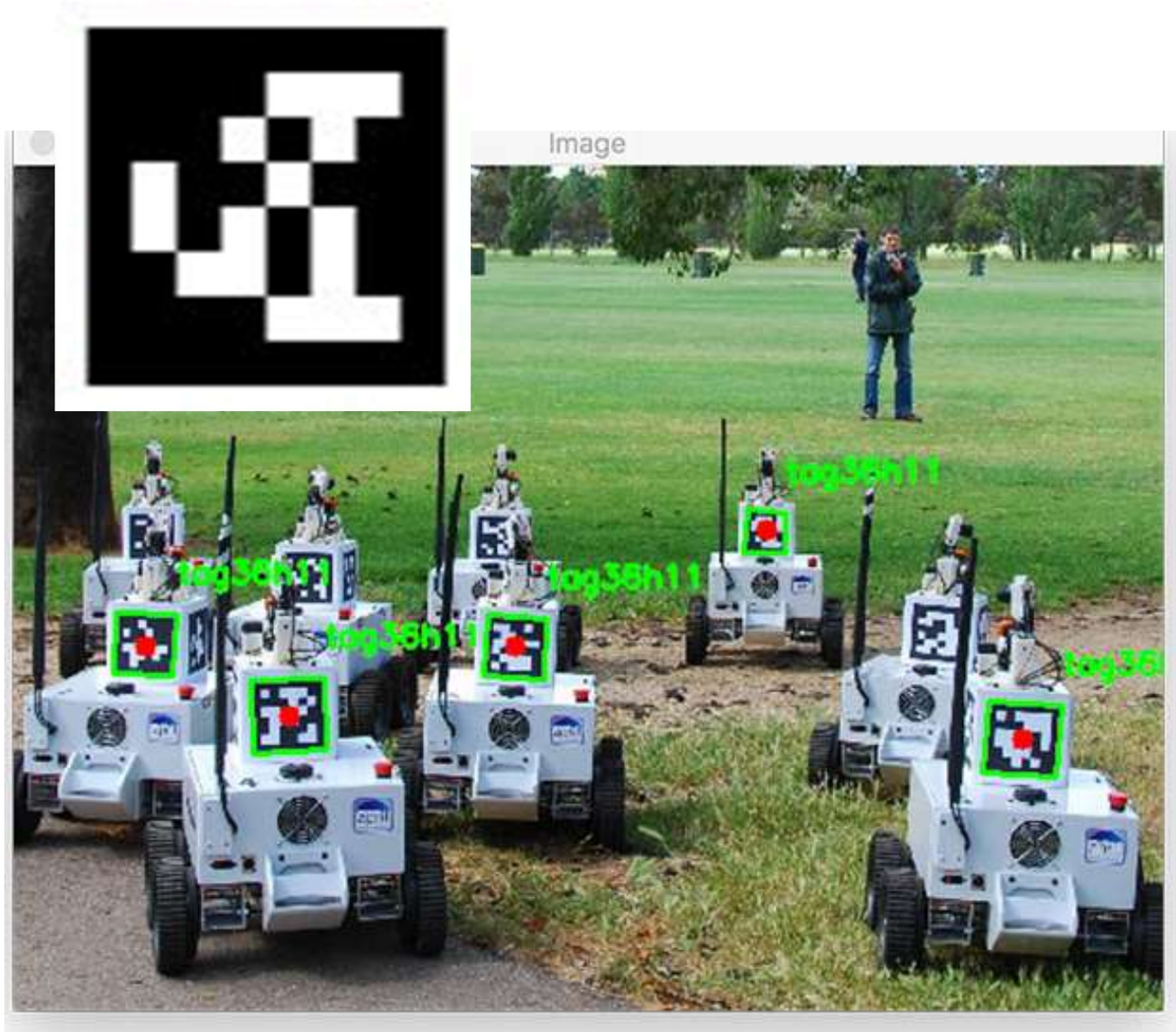
```
pi@raspberrypi:~ $ sudo pip3 install qrcode[pil]
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting qrcode[pil]
  Downloading https://files.pythonhosted.org/packages/42/87/4a3a77e59ab7493d64da
1f69bf1c2e899a4cf81e51b2baa855e8cc8115be/qrcode-6.1-py2.py3-none-any.whl
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from qrcod
e[pil]) (1.10.0)
Requirement already satisfied: pillow; extra == "pil" in /usr/lib/python3/dist-p
ackages (from qrcode[pil]) (4.0.0)
Installing collected packages: qrcode
Successfully installed qrcode-6.1
WARNING: You are using pip version 19.1.1, however version 20.0.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
pi@raspberrypi:~ $
```

QR Code: History

- 1960's: cash registers at supermarkets introduce barcodes
- Barcode are 1-dimensional, stores ~20 alphanumeric characters
- 1990's: QR codes provide 2-dimensional storage of information
  - QR = "Quick Response"
- Wide range of applications, for example:
  1. Mobile applications
  2. Construction sites
  3. Payment
  4. Wifi access
  5. Loyalty programs

AprilTags

- University of Michigan
- 4-12 bits of data only



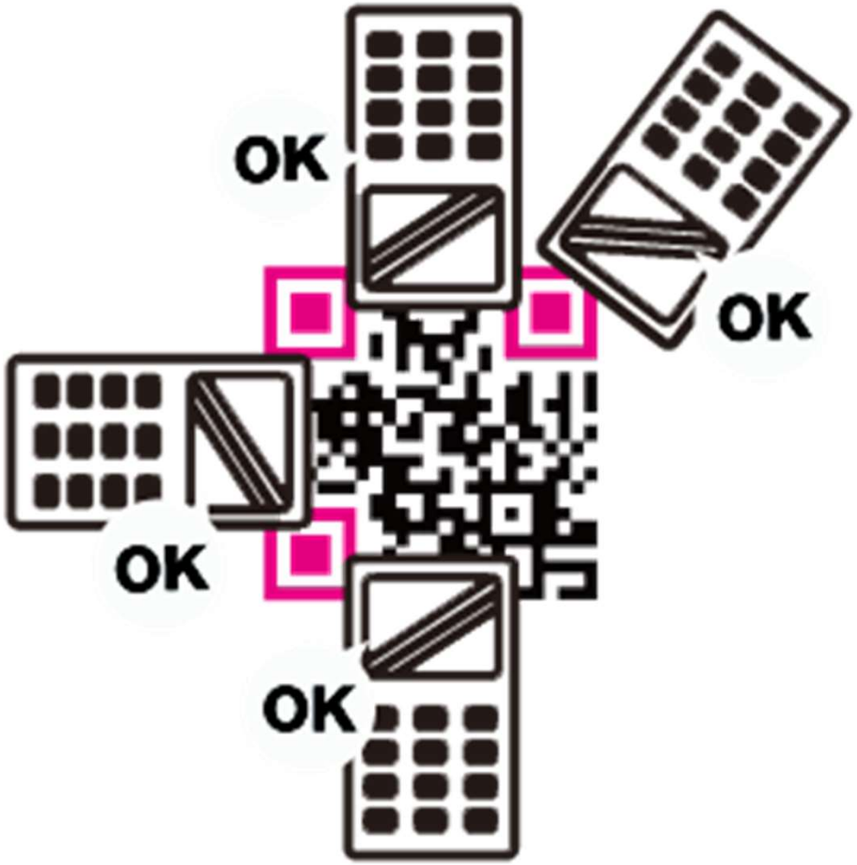
https://www.pyimagesearch.com/2020/11/02/apriltag-with-python/

https://april.eecs.umich.edu/software/apriltag

QR Code: Design

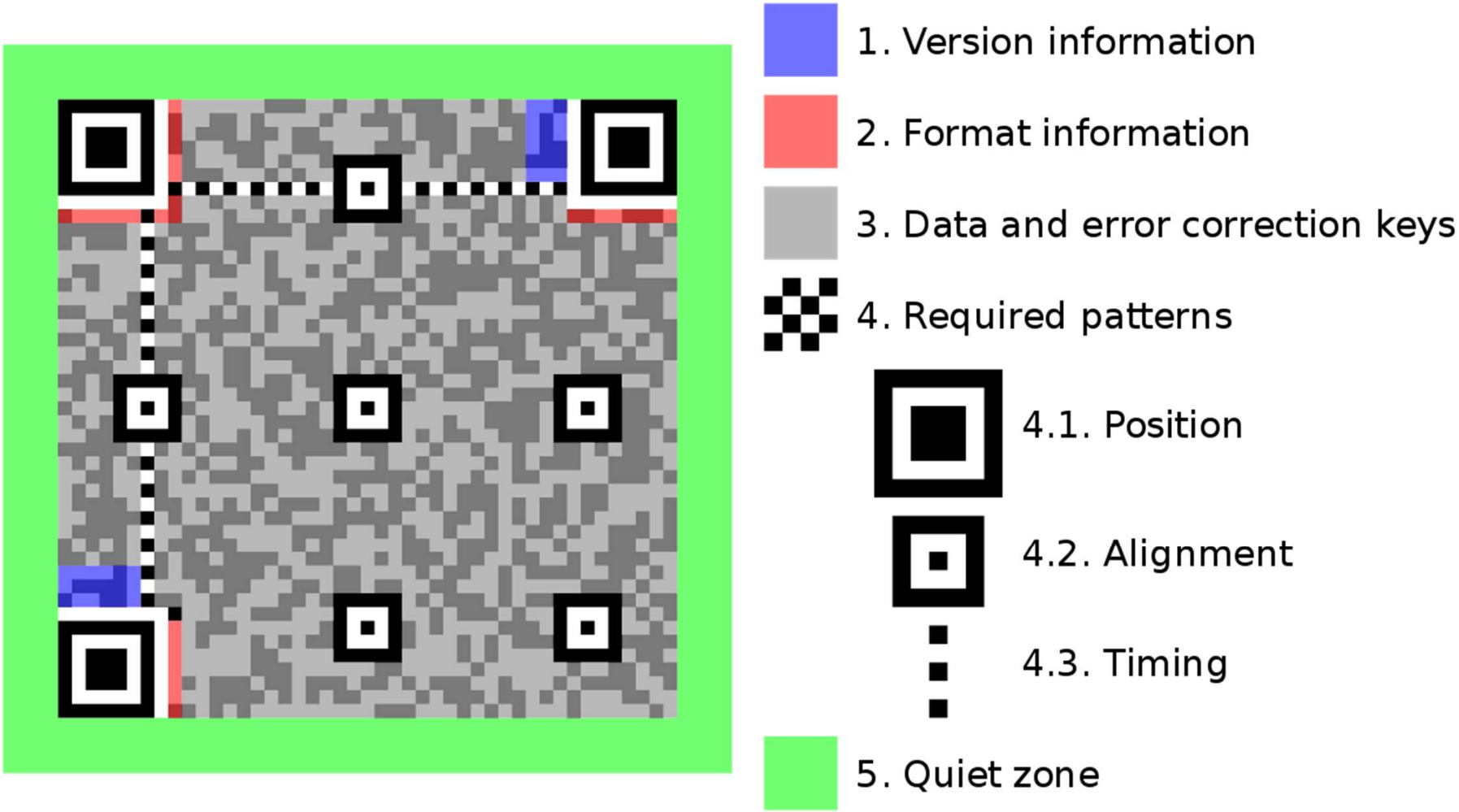
- 1-dimensional barcodes mechanically scanned by a narrow beam of light
- QR codes detected by 2-dimensional image sensor then analyzed by processor
- Locates 3 squares at corners
- Small dots converted to binary numbers

QR Code: Structure

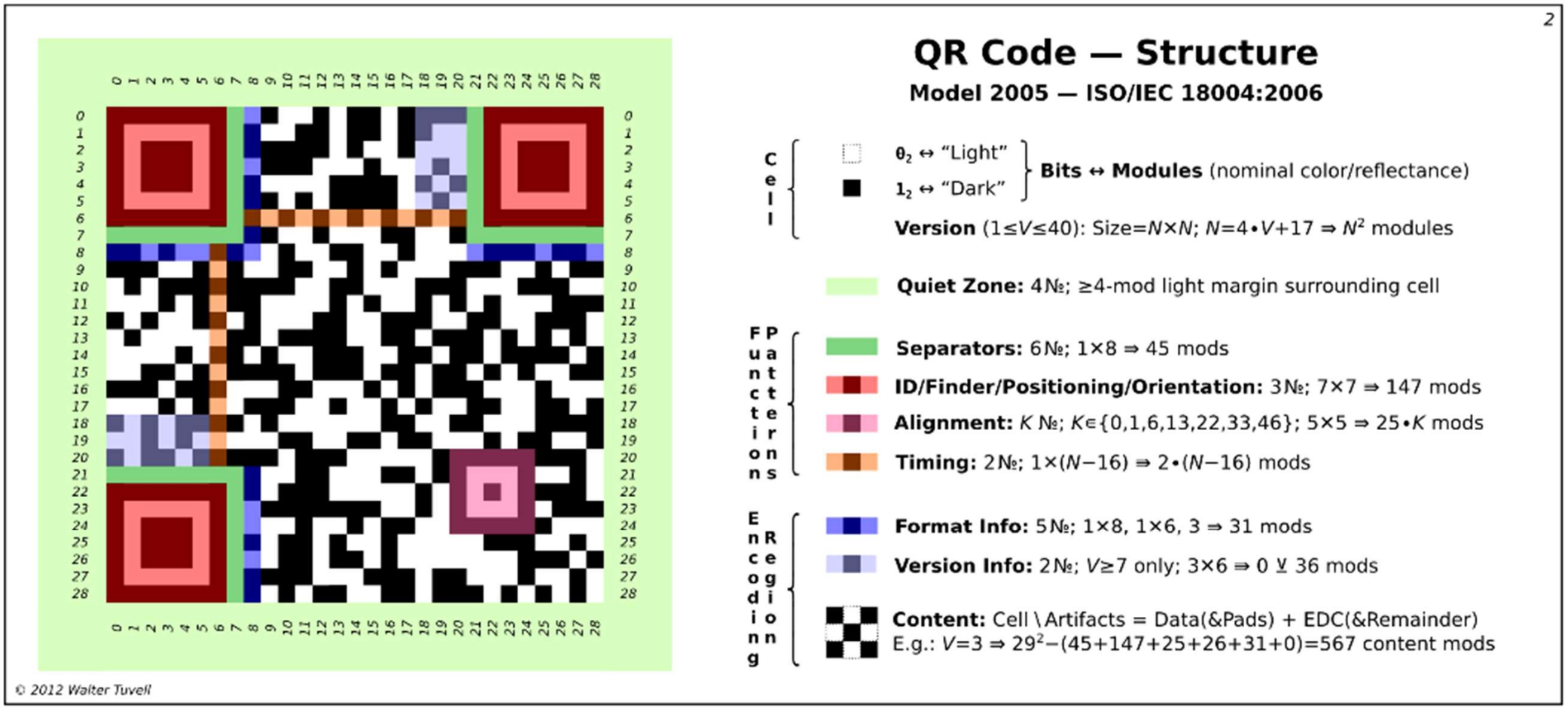


https://www.qrcode.com/en/history/

QR Code: Structure

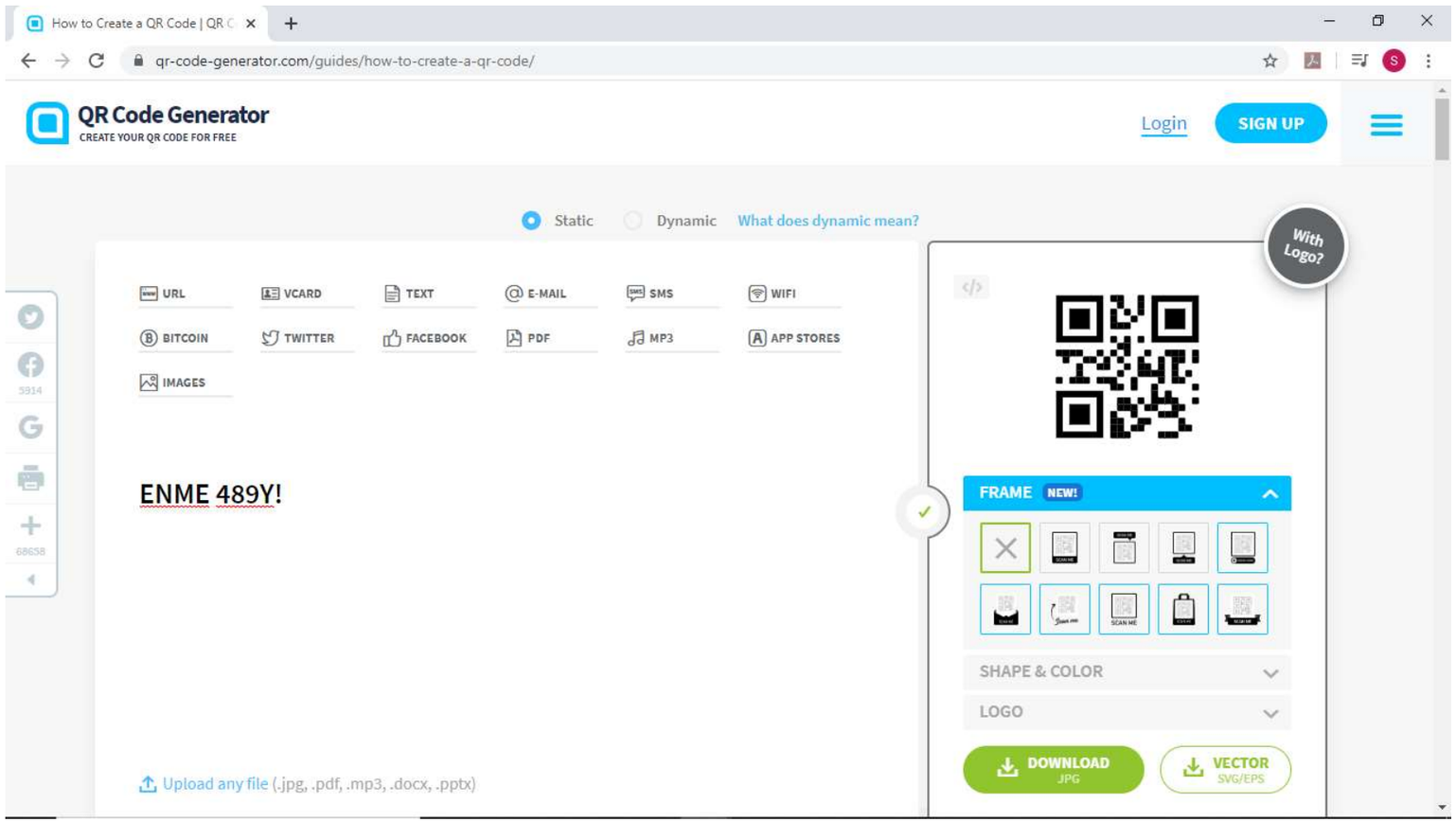


QR Code: Structure



Generate

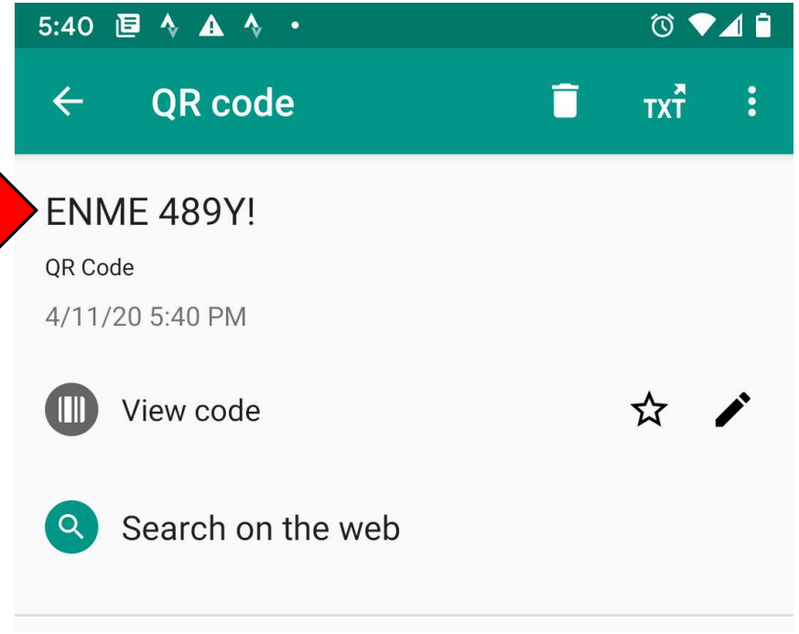
- Online resources



https://www.qr-code-generator.com/guides/how-to-create-a-qr-code/

Generate

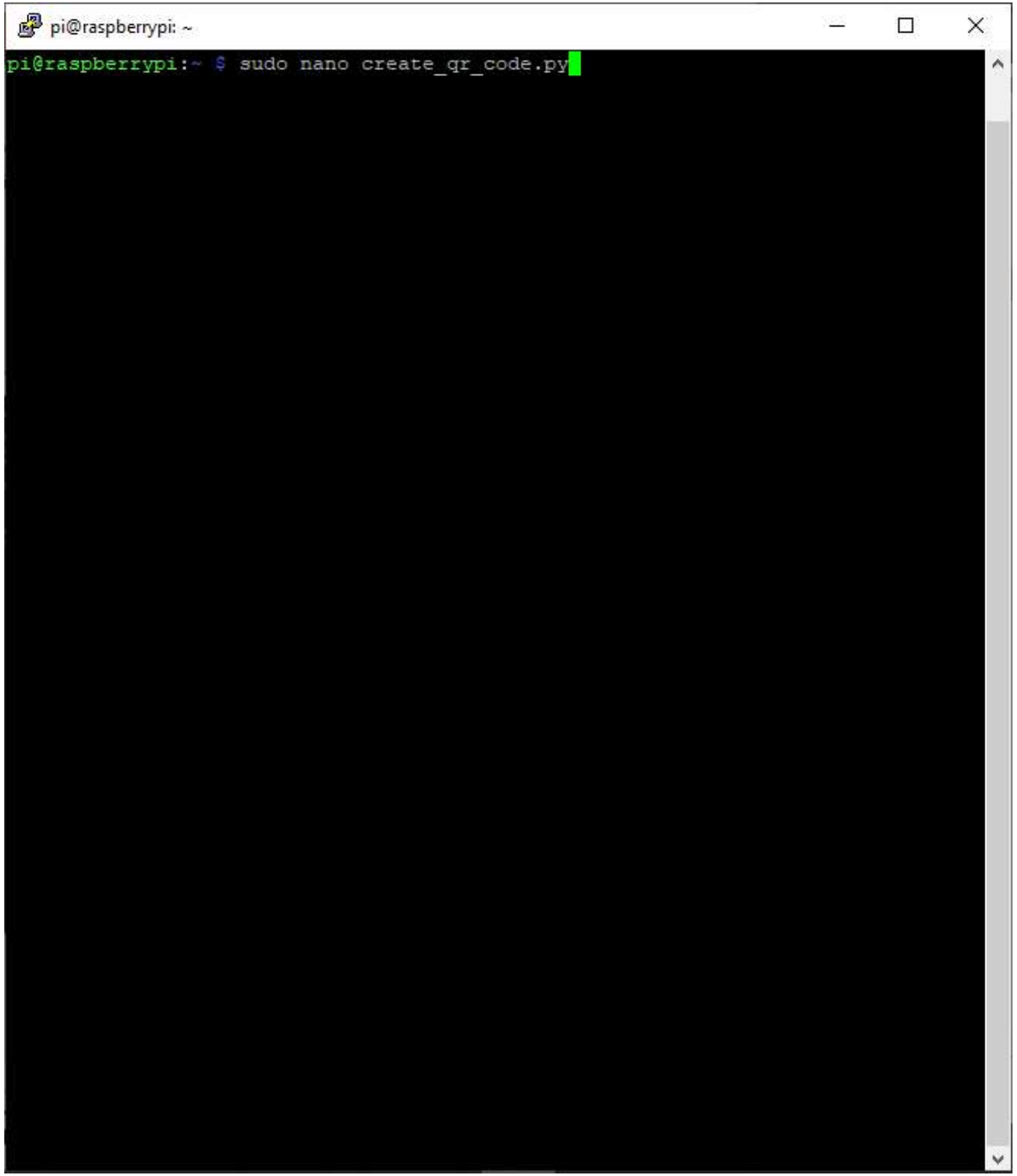
- Online resources



https://www.qr-code-generator.com/guides/how-to-create-a-qr-code/

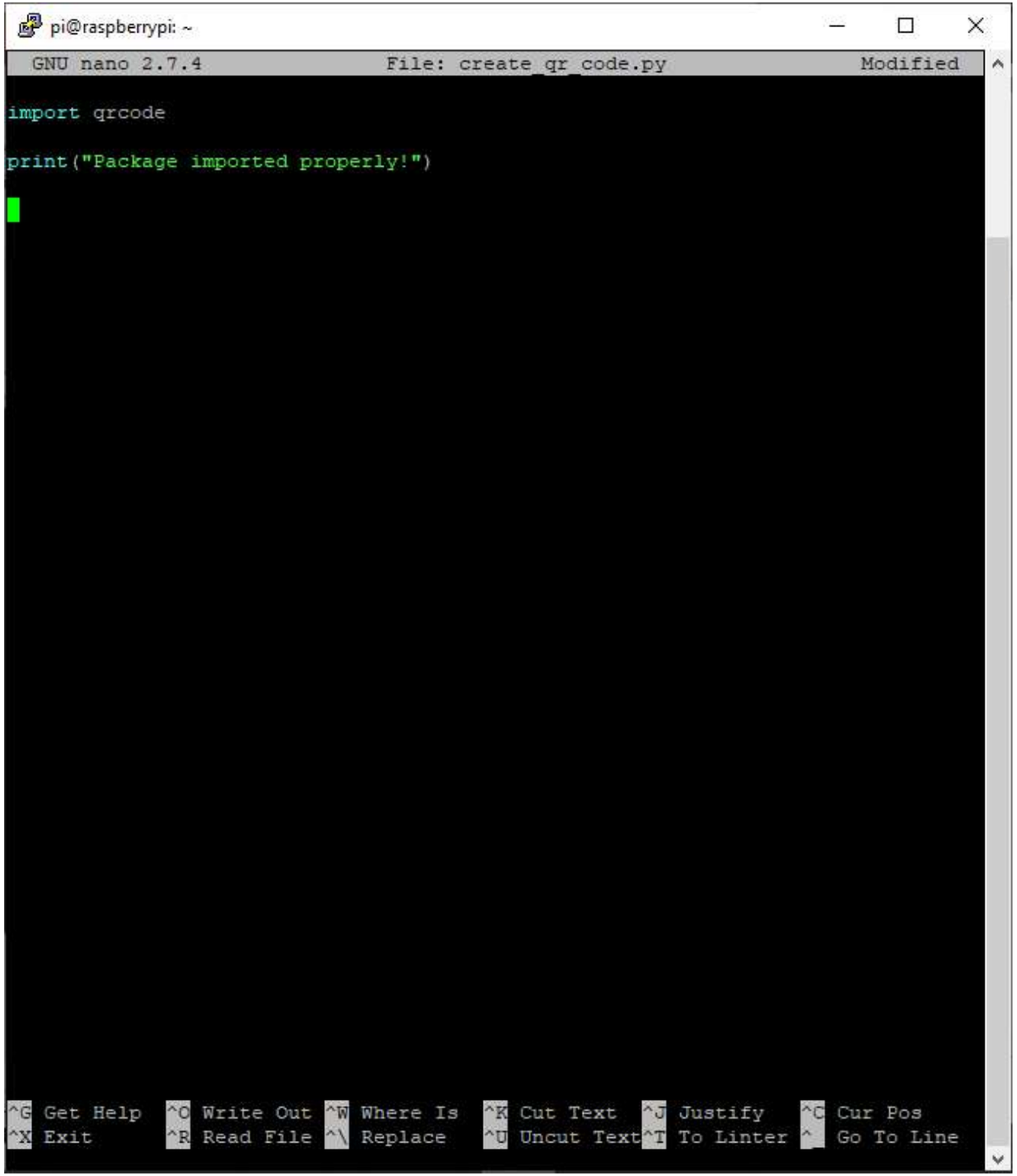
Generate

- Using Python
- Create test script ***create_qr_code.py***



Generate

- Import **qrcode** package



Generate

- Verify qrcode package properly imports

pi@raspberrypi: ~

```
pi@raspberrypi:~ $ sudo nano create_qr_code.py
pi@raspberrypi:~ $ python3 create_qr_code.py
Package imported properly!
pi@raspberrypi:~ $
```

Generate

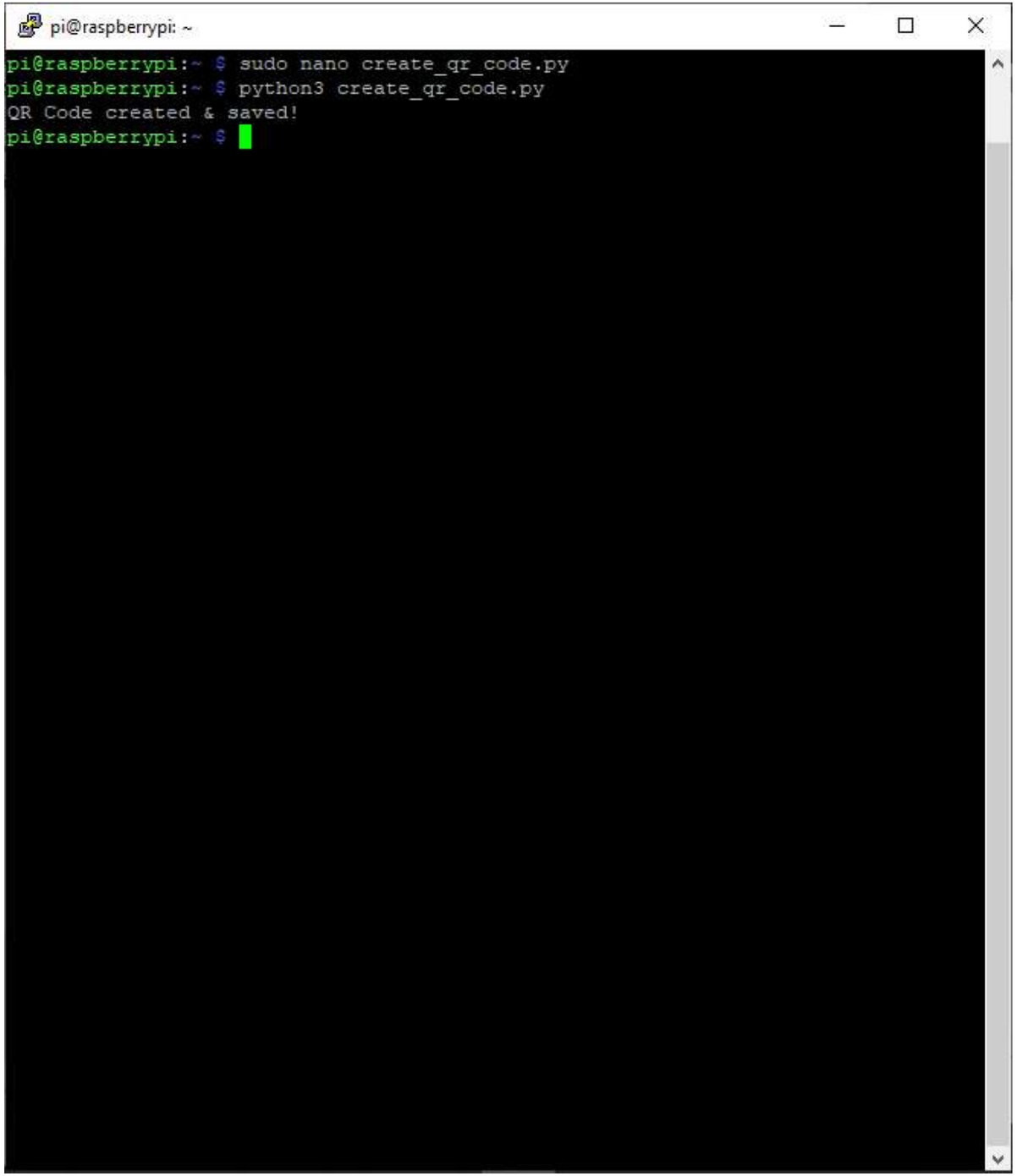
- Create QR Code & save to file

pi@raspberrypi: ~

```
GNU nano 3.2                    create_qr_code.py

import qrcode

code = qrcode.make("ENME 489Y!")

code.save("enme489Yqrcode.png")

print("QR Code generated & saved!")

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

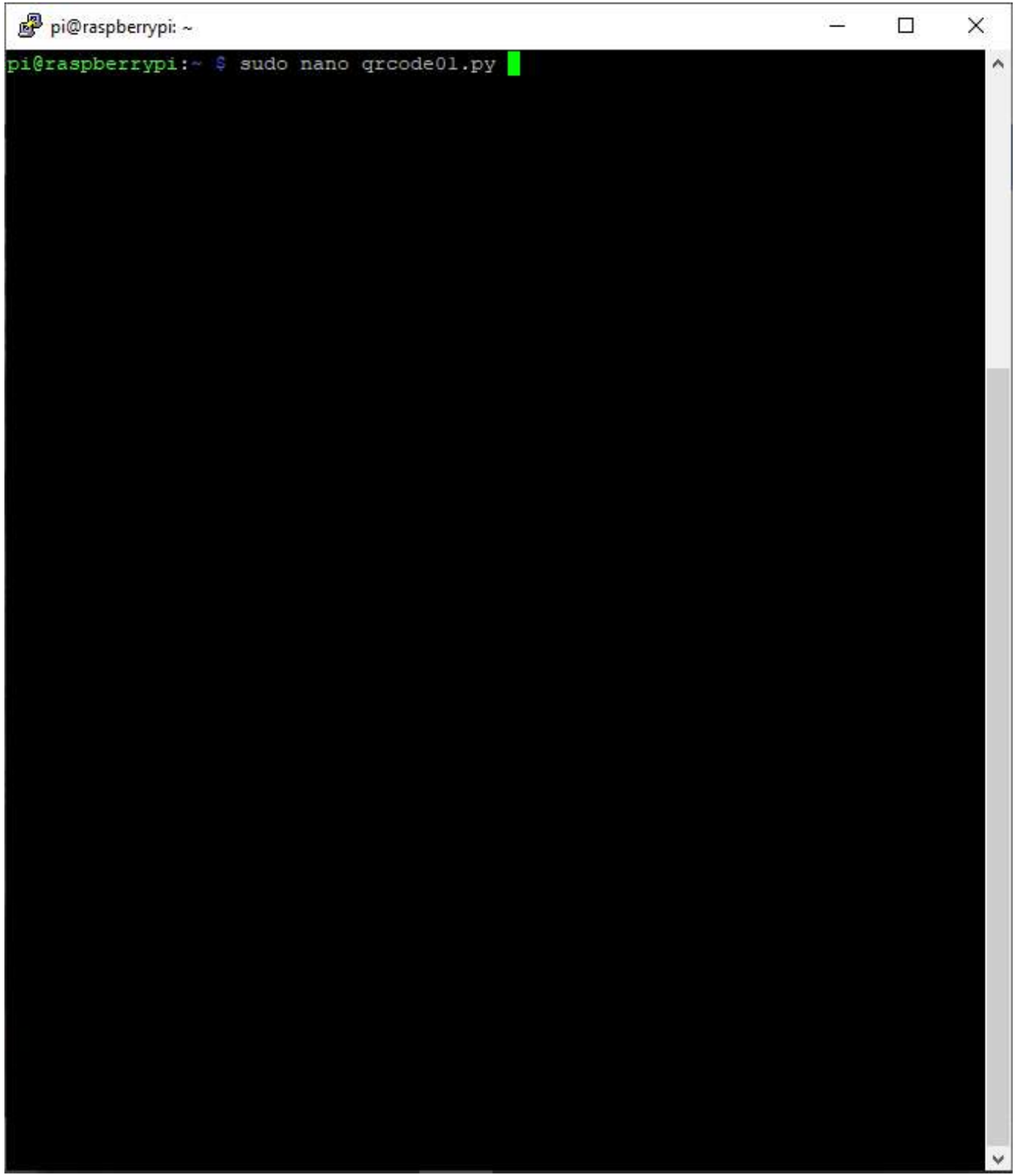
Generate

- Confirm proper information embedded into QR Code



Raspberry Pi

- Create script ***qrcode01.py***

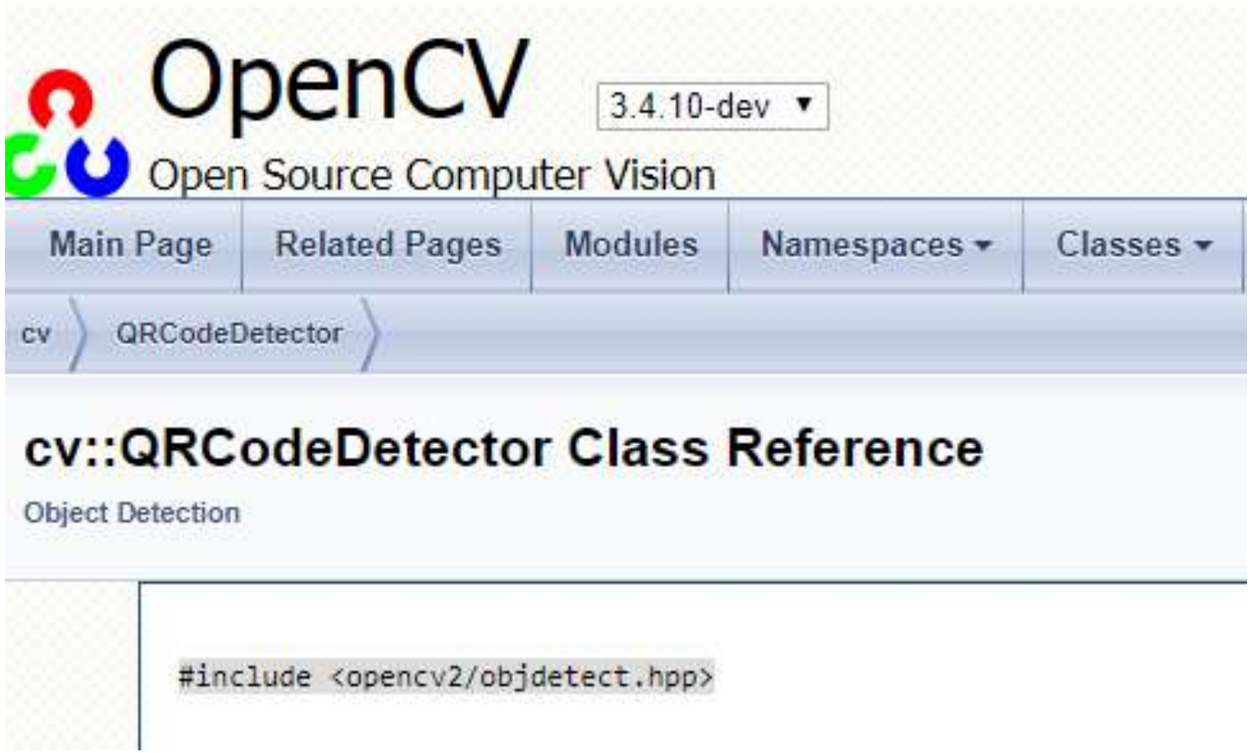


pi@raspberrypi: ~

```
GNU nano 2.7.4                                File: qrcode01.py

import cv2
import os

command = 'sudo modprobe bcm2835-v4l2'
os.system(command)


# Open video capture
cap = cv2.VideoCapture(0)

# Define detector
detector = cv2.QRCodeDetector()

while True:

        check, img = cap.read()

        data, bbox, _ = detector.detectAndDecode(img)

        if(bbox is not None):
                for i in range(len(bbox)):
                        cv2.line(img, tuple(bbox[i][0]), tuple(bbox[(i+1) % len(bbox)][0]), color = (0, 0, 255), thickness = 4)
                        cv2.putText(img, data, (int(bbox[0][0][0]), int(bbox[0][0][1]) - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 0, 0), 2)

        if data:
                print("Data: ", data)

        # Show result to the screen
        cv2.imshow("QR Code detector", img)

        # Break out of loop by presssing the q key
        if(cv2.waitKey(1) == ord("q")):
                break

cap.release()
cv2.destroyAllWindows()

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev Page   M-\ First Line  M-W WhereIs Next
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Linter   ^_ Go To Line  ^V Next Page   M-/ Last Line   M-] To Bracket
```

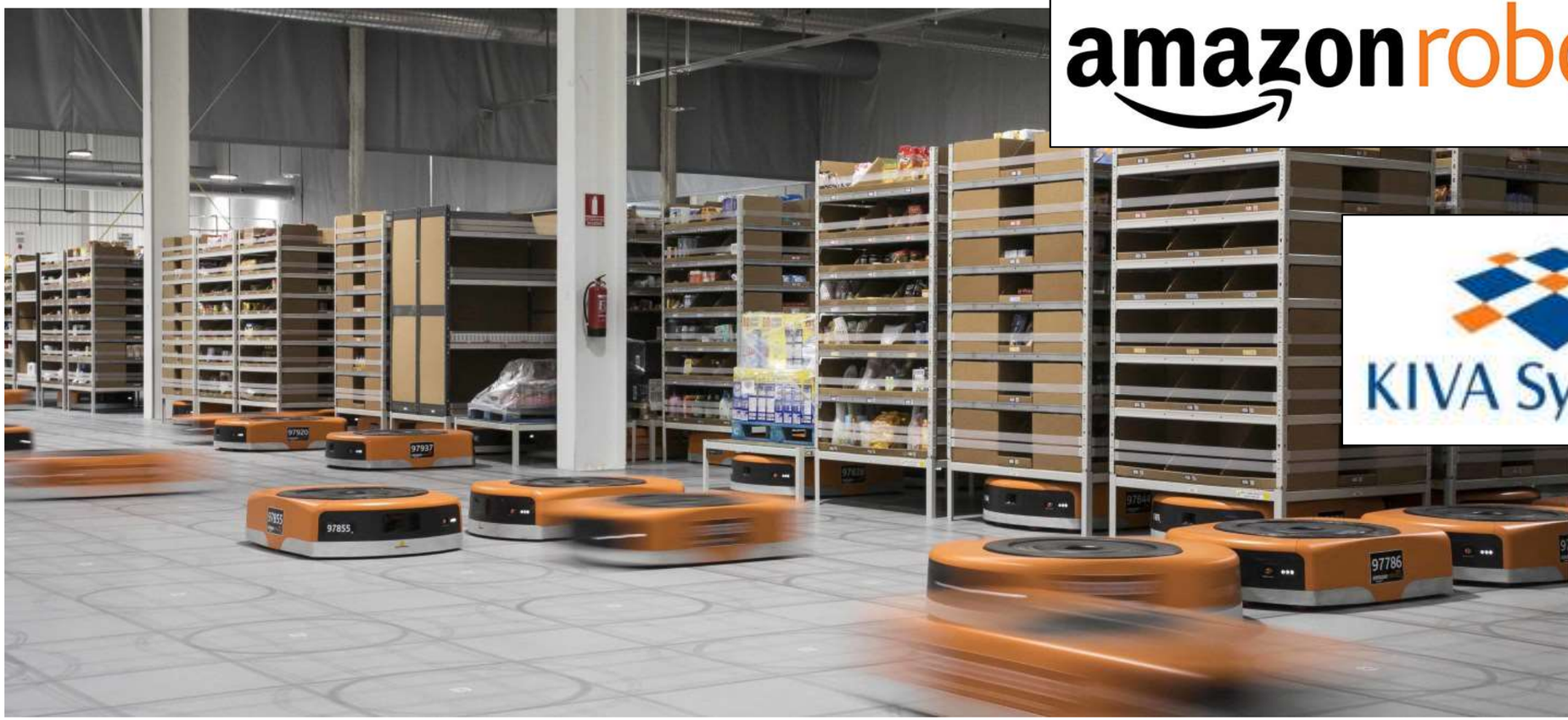
cv2.QRCodeDetector( )

- Available in OpenCV version 4



https://docs.opencv.org/3.4/de/dc3/classcv_1_1QRCodeDetector.html

Case Study: Robotic Control



amazonrobotics

KIVA Systems®

https://www.amazonrobotics.com

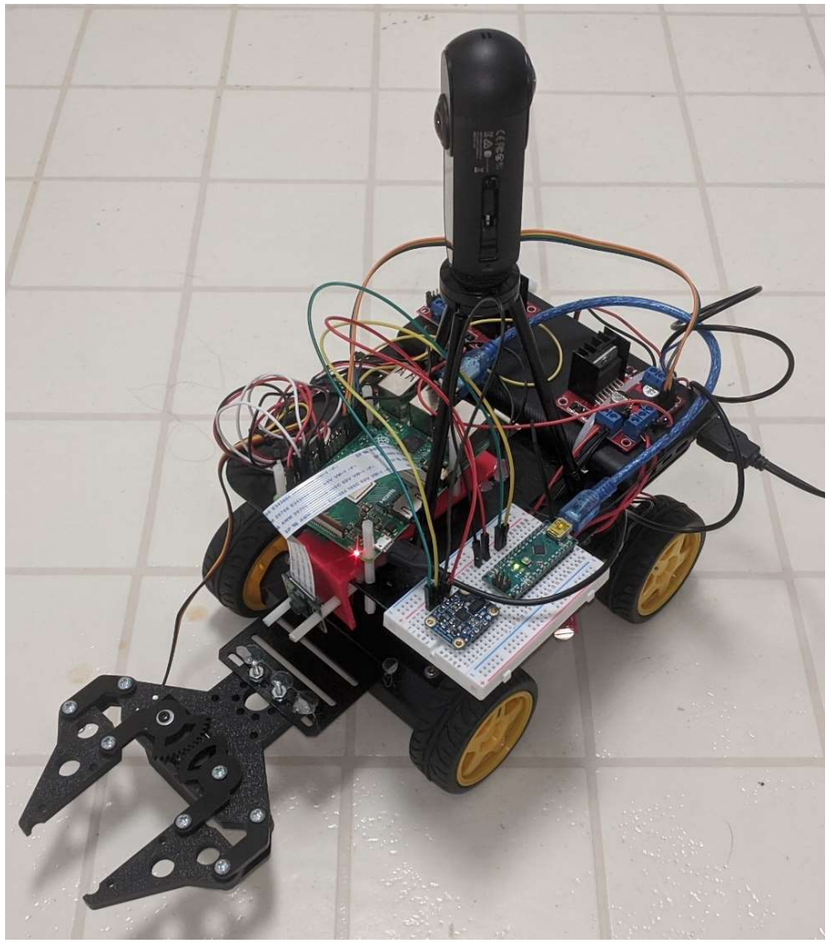
Case Study: Robotic Control

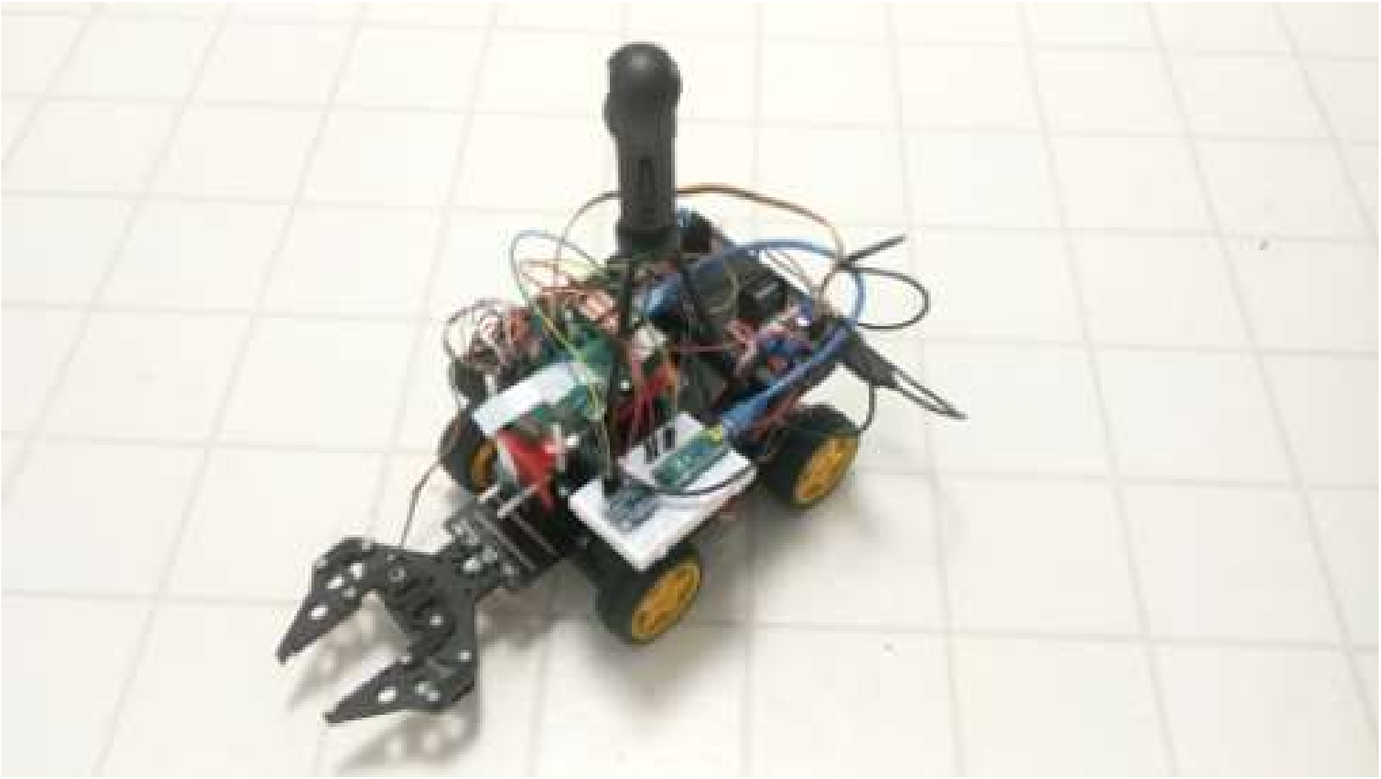
https://www.youtube.com/watch?v=HSA5Bq-1fU4

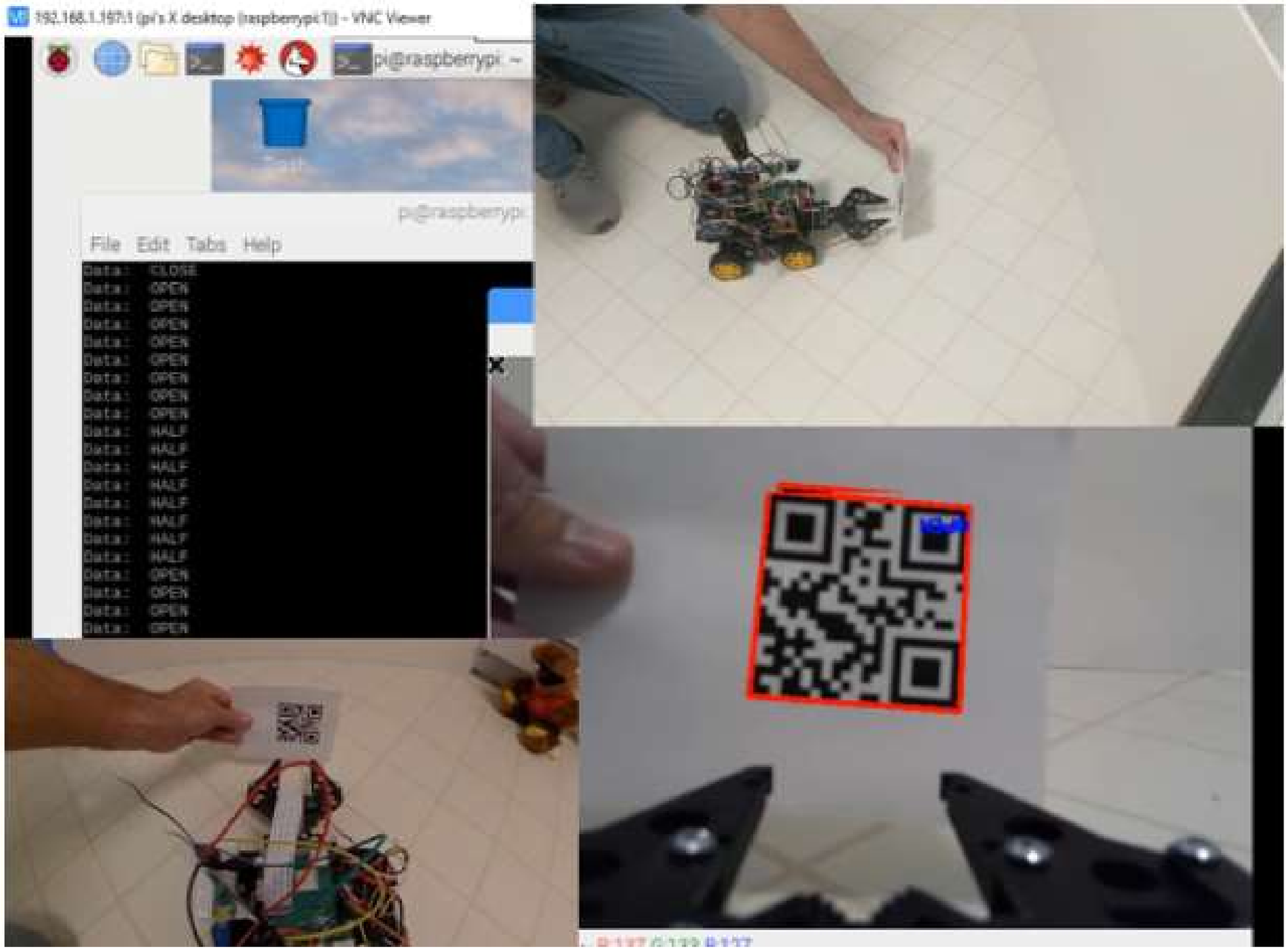
Case Study: Robotic Control

Case Study: Robotic Control

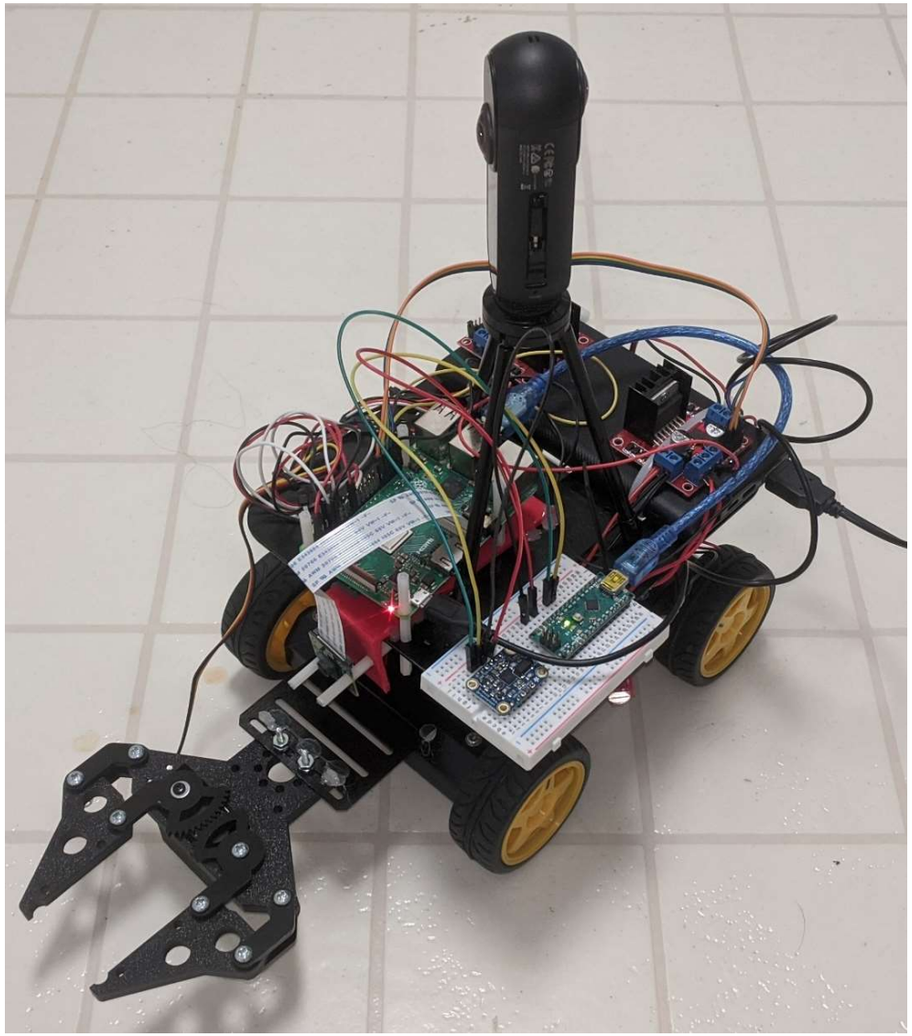
Case Study: Robotic Control






192.168.1.197:1 (pi's X desktop (raspberrypi:1)) - VNC Viewer
pi@raspberrypi: ~
File Edit Tabs Help
Data: CLOSE
Data: OPEN
Data: OPEN
Data: OPEN
Data: OPEN
Data: OPEN
Data: OPEN
Data: OPEN
Data: OPEN
Data: HALF
Data: HALF
Data: HALF
Data: HALF
Data: HALF
Data: HALF
Data: HALF
Data: HALF
Data: OPEN
Data: OPEN
Data: OPEN
Data: OPEN

GPIO



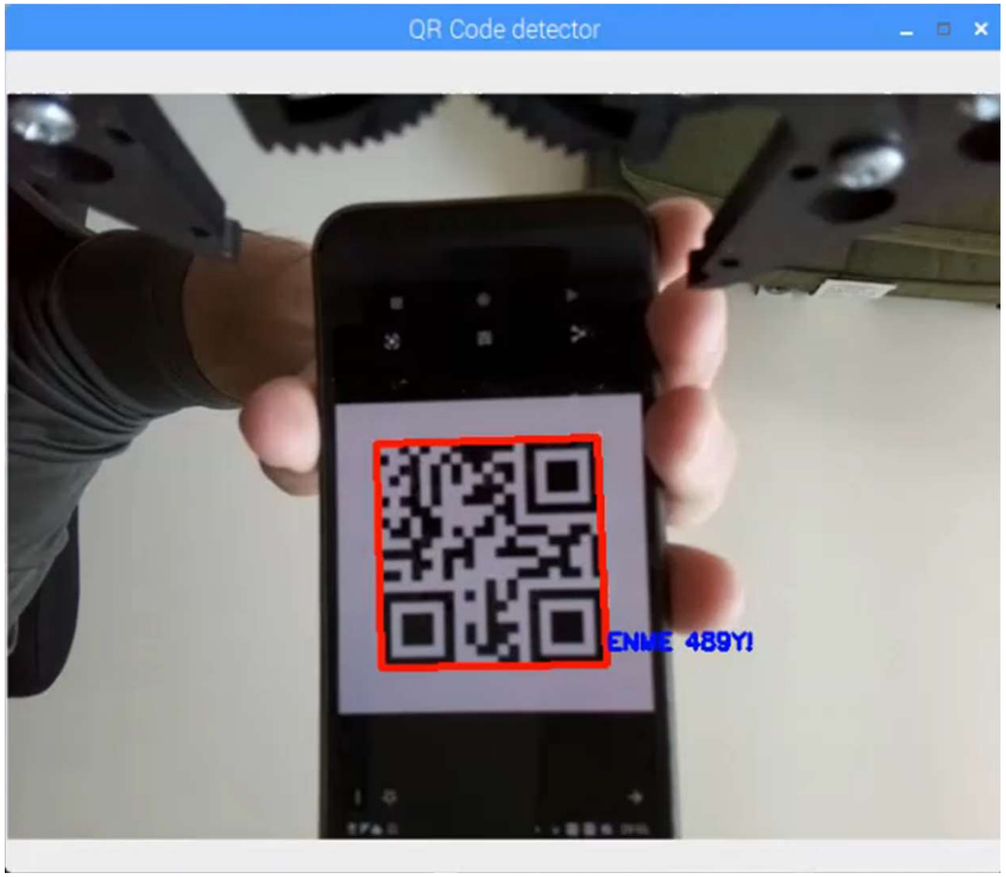
ACT
PWR
GPIO
Raspberry Pi 2 Model B V1.1
© Raspberry Pi 2014

ARDUINO
UNO
A0
A1
A2
A3
A4
A5

In-Class Activity

- Demonstrate full functionality of your QR Code Scanner

1. Brief overview of setup & code
2. **Full demonstration** of capabilities including:
   1. Detection
   2. Decode information
   3. Display information



References

- *Scan QR Codes in Real Time with Raspberry Pi*
  - https://www.hackster.io/gatoninja236/scan-qr-codes-in-real-time-with-raspberry-pi-a5268b?fbclid=IwAR06Kox32mPLqFqtcNvhg2tevcbnf2SRIfh1Kkd9xmJRZqkWFeVl3sVnWGw
- *History of QR Code*
  - https://www.qrcode.com/en/history/
- *QR Code*
  - https://en.wikipedia.org/wiki/QR_code
- *QR Code Generator*
  - https://www.qr-code-generator.com/guides/how-to-create-a-qr-code/
- *Future of Work Automation – 3.3: Amazon's Warehouses Robots / Machines*
  - https://medium.com/@msjulieho/future-of-work-automation-amazons-warehouses-robots-machines-51338419c89d
- *Meet the Robots at Amazon*
  - https://www.youtube.com/watch?v=HSA5Bq-1fU4
- *Amazon's $775 million deal for robotics company Kiva is starting to look really smart*
  - https://www.businessinsider.com/kiva-robots-save-money-for-amazon-2016-6