2. QR code

2.1. Introduction to QR code

QR code is a type of two-dimensional barcode. QR comes from the abbreviation of "Quick Response" in English, which means quick response. It comes from the inventor's hope that QR code can allow its content to be decoded quickly. QR code not only has large information capacity, high reliability, and low cost, but also can express a variety of text information such as Chinese characters and images. It has strong confidentiality and anti-counterfeiting properties and is very convenient to use. More importantly, the QR code technology is open source.

2.2. Structure of QR code

picture	Parse
	**Positioning** markings indicate the direction of the QR code.
	Alignment markings If the QR code is large, these additional elements help with positioning.
	**pattern** With these lines, the scanner can identify how big the matrix is.
	**Version information** (Version information) here specifies the version number of the QR code in use. There are currently 40 different version numbers of the QR code. Version numbers for the sales industry are usually 1-7.
	**Format** information Format patterns contain information about fault tolerance and data mask patterns and make scanning codes easier.
	**Data** and error correction keys These modes hold the actual data.
	**Quiet** zone This zone is very important for the scanner, its role is to separate itself from the surrounding.

2.3. Characteristics of QR code

The data value in the QR code contains repeated information (redundant value). Therefore, even if up to 30% of the QR code structure is destroyed, it does not affect the readability of the QR code. The storage space of the QR code is up to 7089 bits or 4296 characters, including punctuation and special characters, which can be written into the QR code. In addition to numbers and characters, words and phrases (such as URLs) can also be encoded. As more data is added to the QR code, the code size increases and the code structure becomes more complex.

2.4. QR code creation and recognition

Install the relevant environment (the supporting virtual machine has already set up the environment)

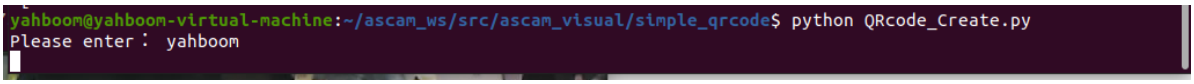
```
sudo python3 -m pip install qrcode pyzbar
sudo apt-get install libzbar-dev
```

- Create

Source code location: ~/ascam_ws/src/ascam_visual/simple_qrcode/QRcode_Create.py

```
cd ~/ascam_ws/src/ascam_visual/simple_qrcode
python QRcode_Create.py
```

Enter the content to be generated in the terminal and press Enter to confirm.

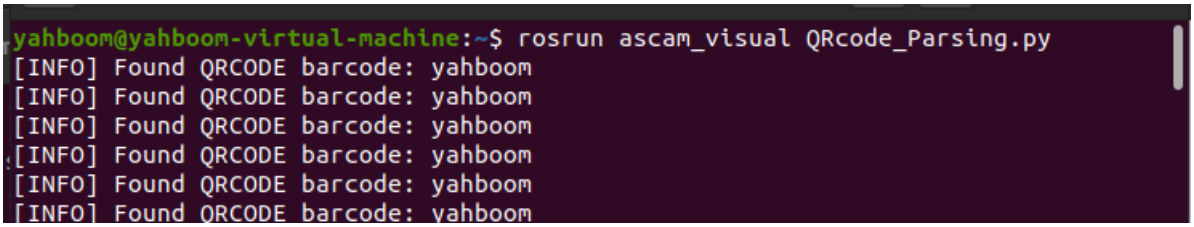


- Recognition

Source code location: ~/ascam_ws/src/ascam_visual/simple_qrcode/QRcode_Parsing.py

Start the camera

```
roslaunch ascamera hp60c.launch
rosrun ascam_visual QRcode_Parsing.py
```



After normal startup, put the QR code in front of the camera. After recognizing the QR code, the QR code will be framed and the QR code content will be printed out.