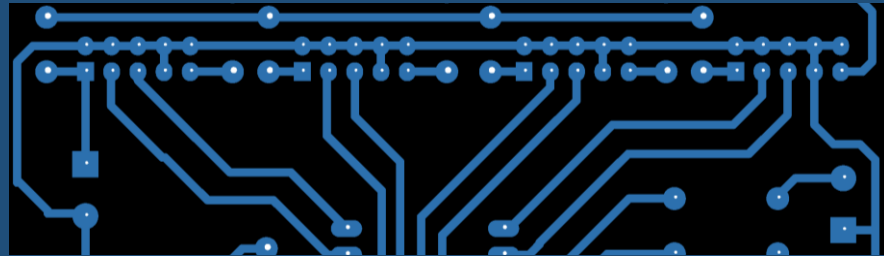


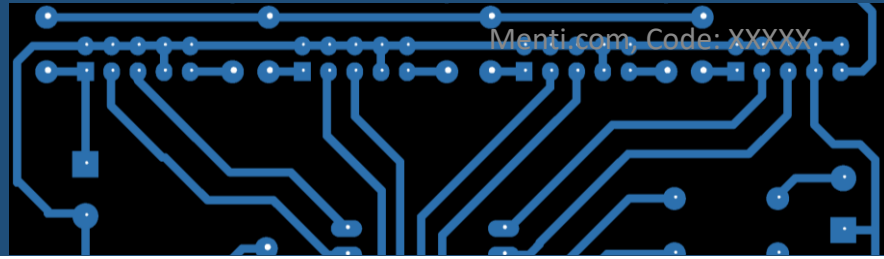
EEEE1039
Applied Electrical and Electronic Engineering: Construction Project

Dr. Shuo Wang
shuo.wang@nottingham.edu.cn

Department of Electrical and Electronic Engineering
Nottingham Ningbo China Beacons of Excellence
Research and Innovation Institute
University of Nottingham Ningbo China

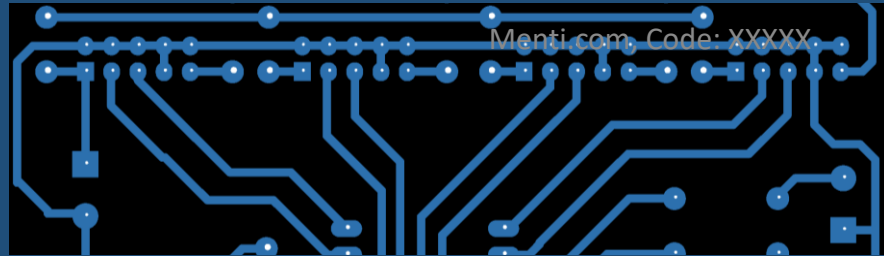
The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

EEEE1039:
Content list



- Project 6 & 7 Introduction
  - General
  - Components and System
  - OpenCV
- Assessments
- Component recycle

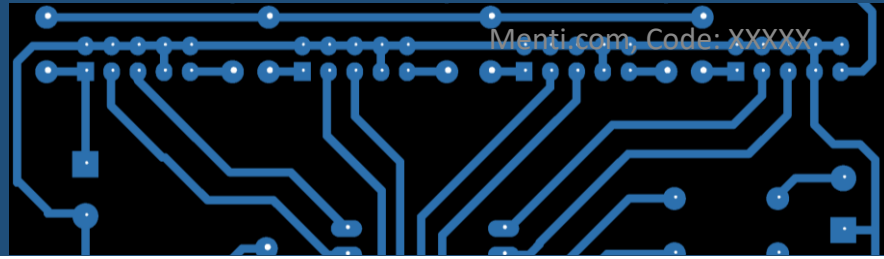
EEEE1039:



- Project 6 & 7 Introduction
  - General
  - Components and System
  - OpenCV
- Assessments
- Component recycle

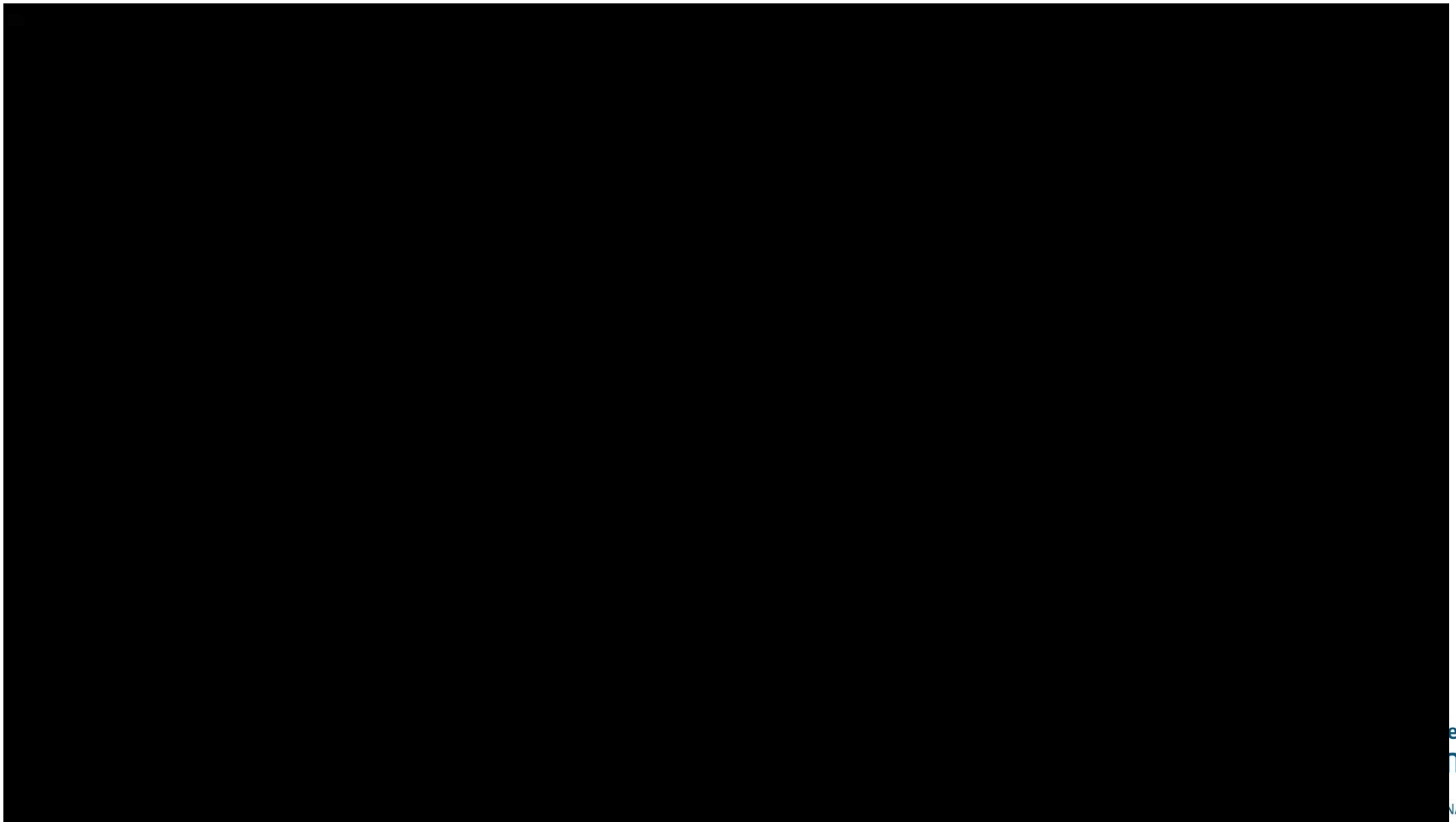
EEEE1039:

Project 6 & 7 – previous year



Video from previous year…

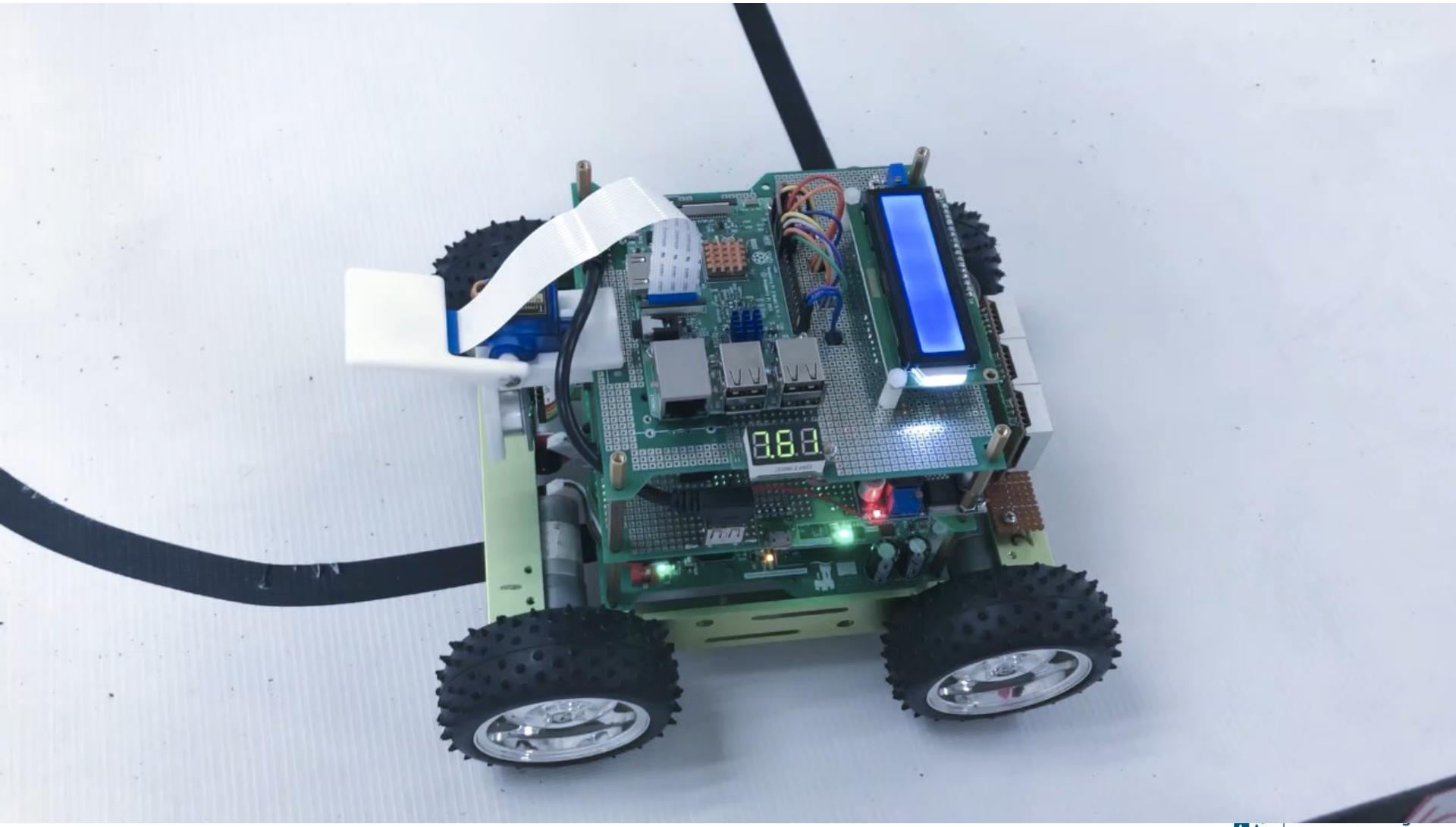
- Short cuts, Kick football, Traffic light



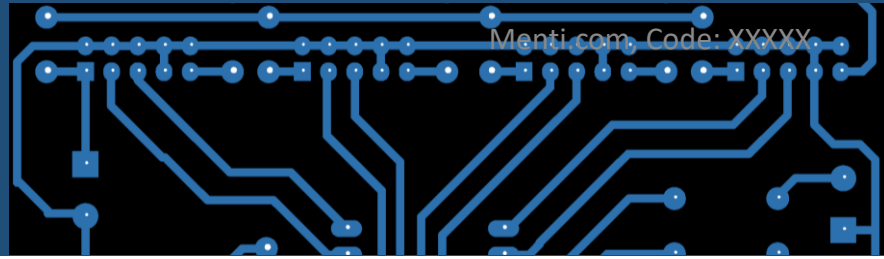
EEEE1039:

Project 6 & 7 – previous year

Menti.com Code: XXXXX



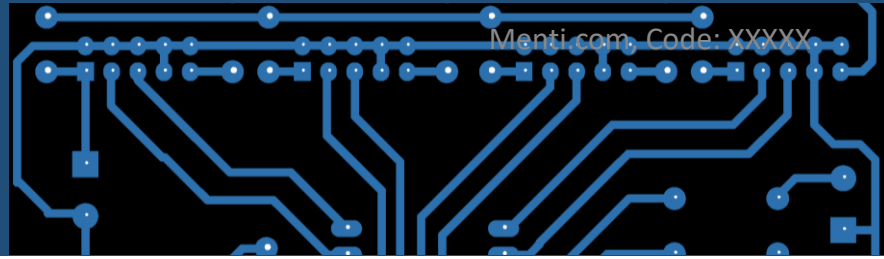
EEEE1039:
Project 6 & 7 – this year



Advanced line following with Object recognition and reaction! Image Processing Required!

- Use C/C++ function libraries: **OpenCV**
- Object recognition
  - Object Tracking: Follow the lead https://www.youtube.com/watch?v=3BJFxnap0AI
  - Object colour recognition https://www.youtube.com/watch?v=hQ-bpfdWQh8
  - Template matching https://www.youtube.com/watch?v=SUU_kNls5Ak
- Ultimately to follow a line – a single line, dual line, black on white, white on black, colour… https://www.youtube.com/watch?v=BBwEF6WBUQs

EEEE1039:
Project 6 & 7 – Tasks



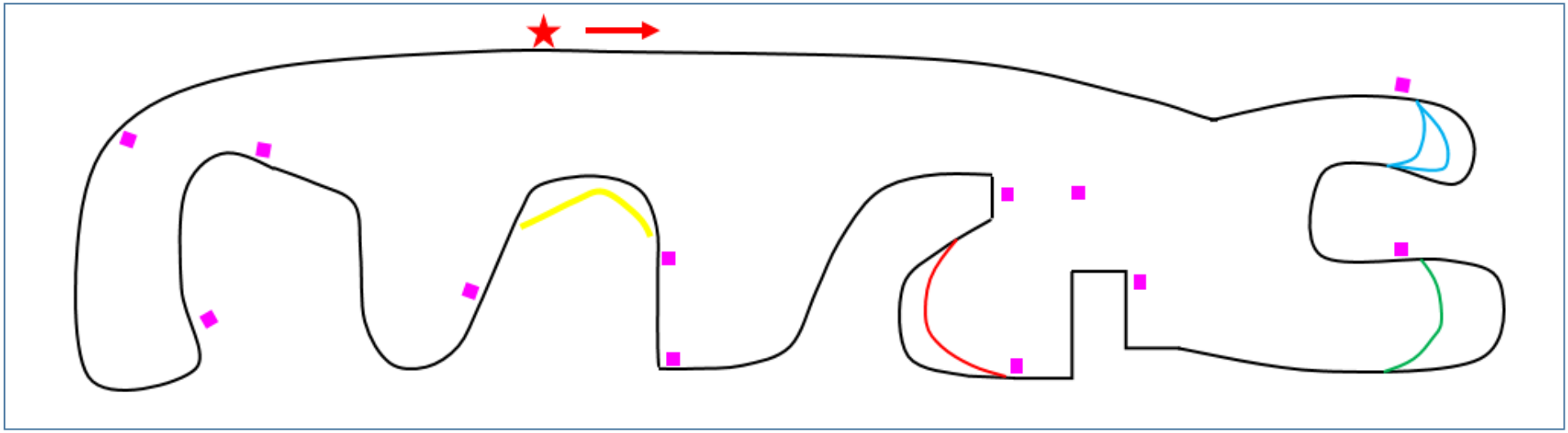
- Set up Raspberry Pi with the car
- Set up hardware with Raspberry Pi
- Assemble and test whole system
- OpenCV tasks
- Demo

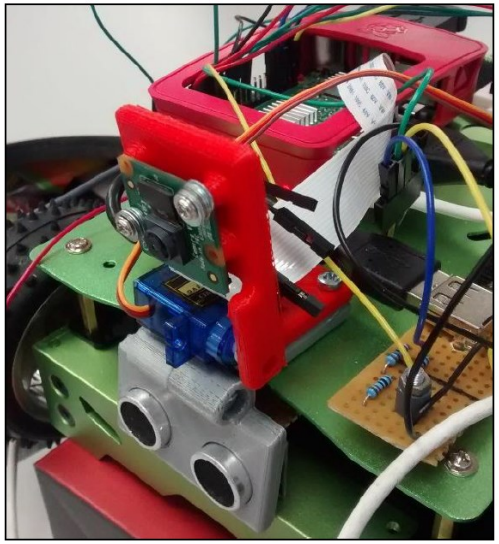
- LCD
- HC-SR04 distance sensor
- Level shifter
- Pi power connector
- Camera
- SG-90 servo motor
- Audio power amplifier

- **Line following**
- Count Shapes
- Short cuts
- Play music
- Alarm flash
- Approach and stop
- Bonus: traffic light
- Bonus: kick football
- Bonus: multiple tasks

EEEE1039:

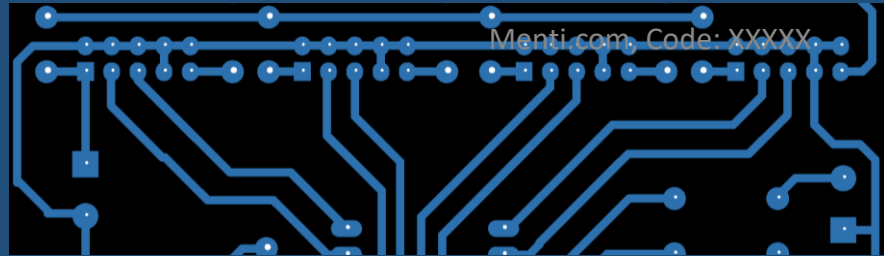
Project 6 & 7 – this year





The track is only a schematic and will be updated depending on actual layout.

EEEE1039:
Content list



- Project 6 & 7 Introduction
  - General
- Components and System
  - OpenCV
- Assessments
- Component recycle

EEEE1039:

Components – what you already know

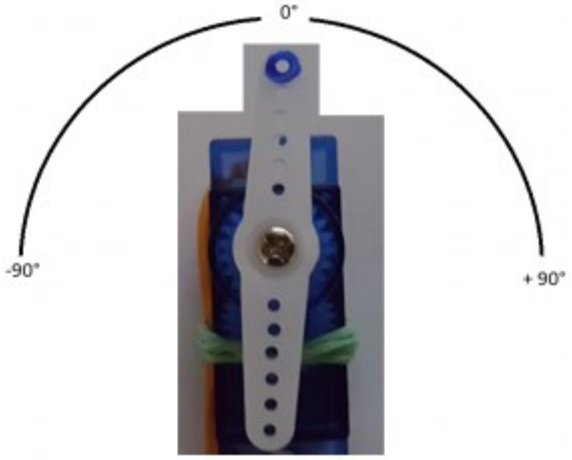
Menti.com Code: XXXXX

- Raspberry Pi 3B
- LCD1602
- HC-SR04 distance sensor
- Level shifter (4-channel)
- Pi power connector
- Camera
- Audio power amplifier
- SG-90 servo motor

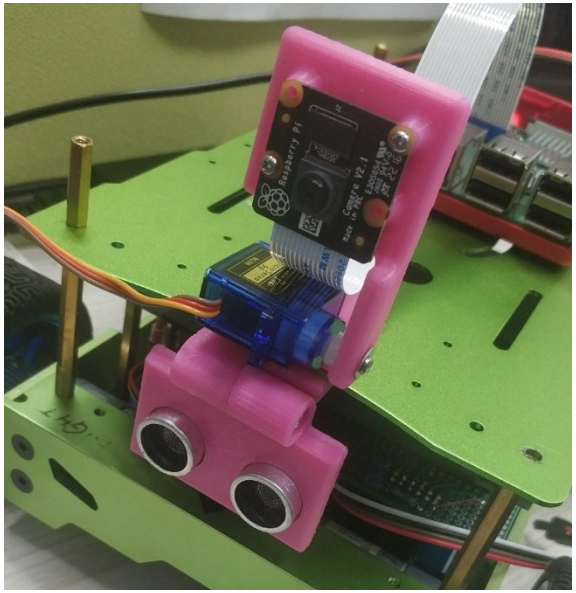
EEEE1039:
Components – Servo motor

Menti.com Code: XXXXX

- Motorized angle control for the camera
  - You may need to move your camera to in between the line following task and the symbol recognition task



SG90 Micro servo



Motorized angle control for the camera

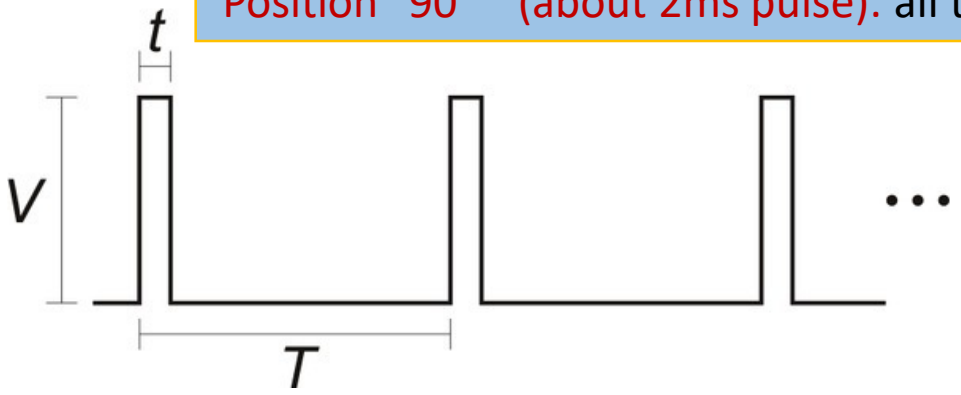
- SG90 specs
  - Rotational Range: 180°
  - Analog Modulation: #include <softPwm.h>
  - Operational Speed: 0.1sec/60°(at 4.8V)

http://wiki.jmoon.co/sg90/

EEEE1039:
Components – Servo motor

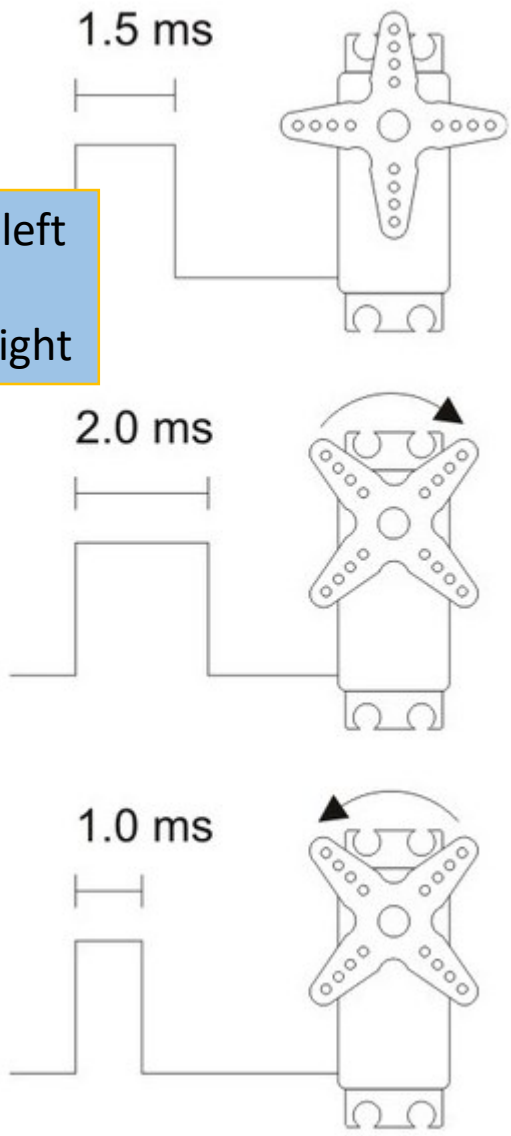
Menti.com, Code: XXXXX

Position "-90^o " (about 1ms pulse): all the way to the left
Position "0^o" (1.5 ms pulse): middle position
Position "90^o " (about 2ms pulse): all the way to the right



- Pulse voltage, V
- Pulse width, t
- Pulse period, T

Pulse wave for the position control of SG90

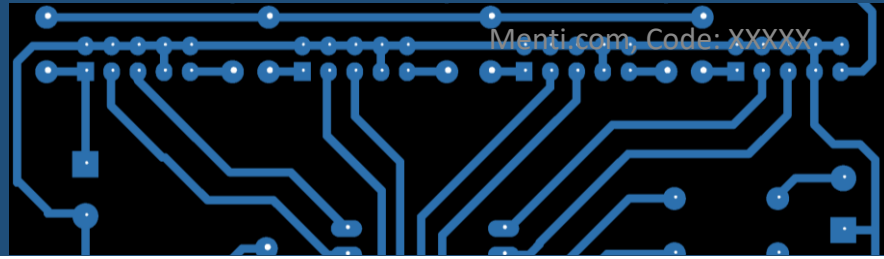


Motorized angle control for the camera

- T = 20ms
- t is from 1.0ms to 2.0ms

https://www.pololu.com/blog/17/servo-control-interface-in-detail

EEEE1039:
The whole system

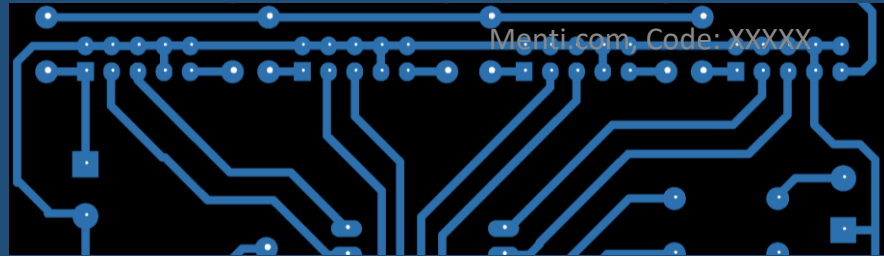


Assemble the whole system and enhance the hardware

Raspberry Pi 26 GPIO, besides VCC and GND:

- 2 for Vehicle, UART
- 6 for LCD1602
- 2 for HC-SR04
- 1 for SG-90
- Voltage level shifters
- Pi Camera
- Audio power amplifier

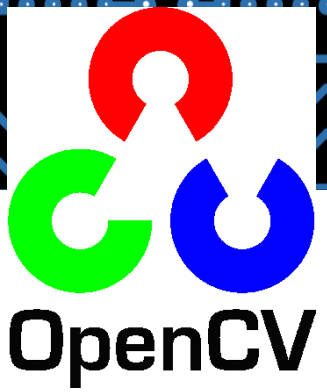
EEEE1039:
Content list



- Project 6 & 7 Introduction
  - General
  - Components and System
  - OpenCV
- Assessments
- Component recycle

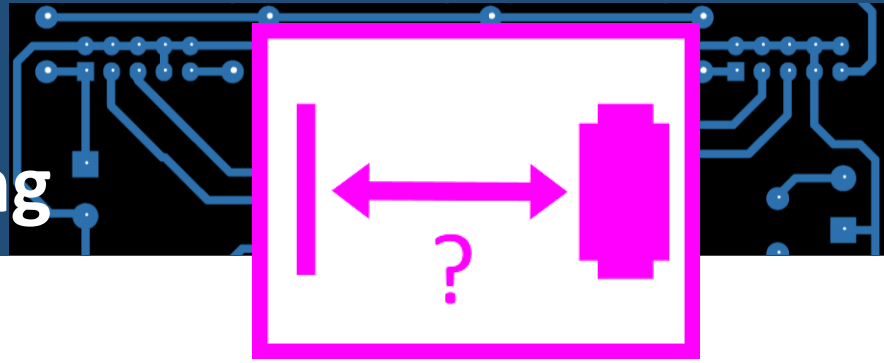
EEEE1039:
OpenCV

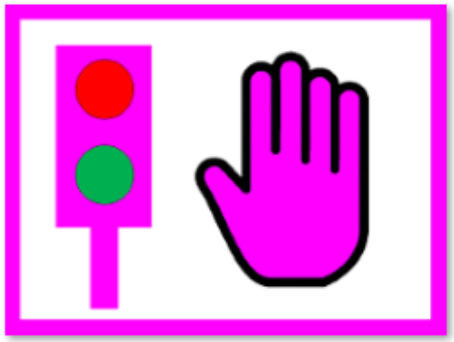
Menti.com Code: XXXXX



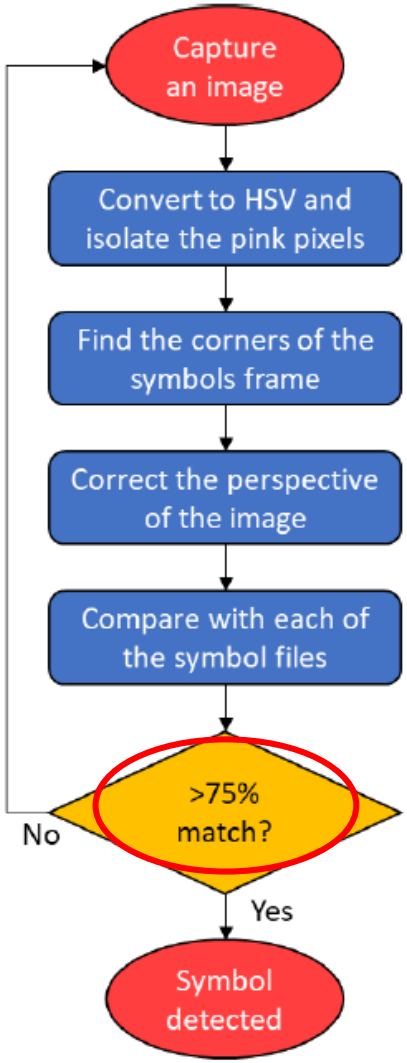
- You know it by now!
- Example code and user-defined libraries are provided in the image under "**OpenCV-AEE**"
- Algorithm: **Template Matching only**
- Different from the coursework: instead of processing a recorded video, process the Pi camera live frames in real time
- General image processing techniques have been introduced in the coursework introduction, refer to it

EEEE1039:
Open CV – Template Matching





Sample symbols

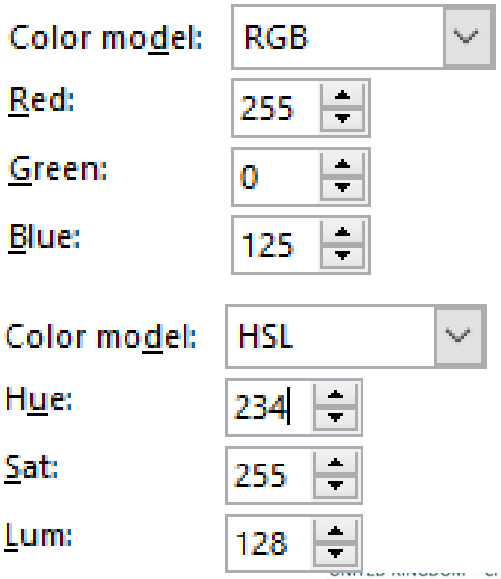


We can use `findContours()` here, but we may need to simplify the contour to get four corners

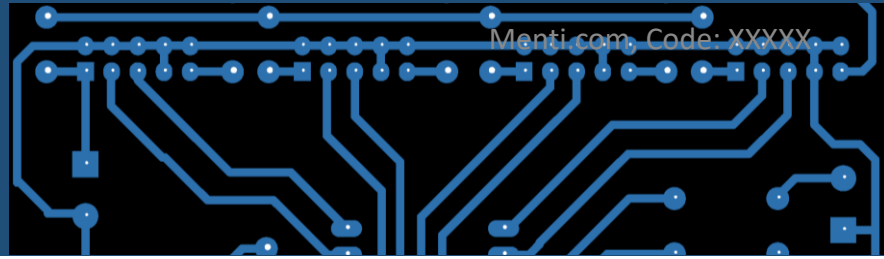
This is achieved with `transformPerspective()`

The symbol and image must be black and white, then `compareImages()` can be used

Flow chart for symbol identification



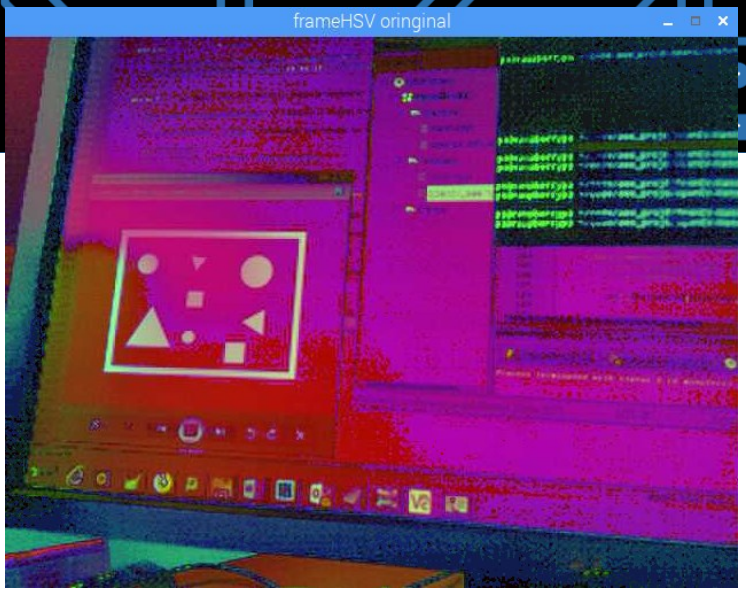
EEEE1039:

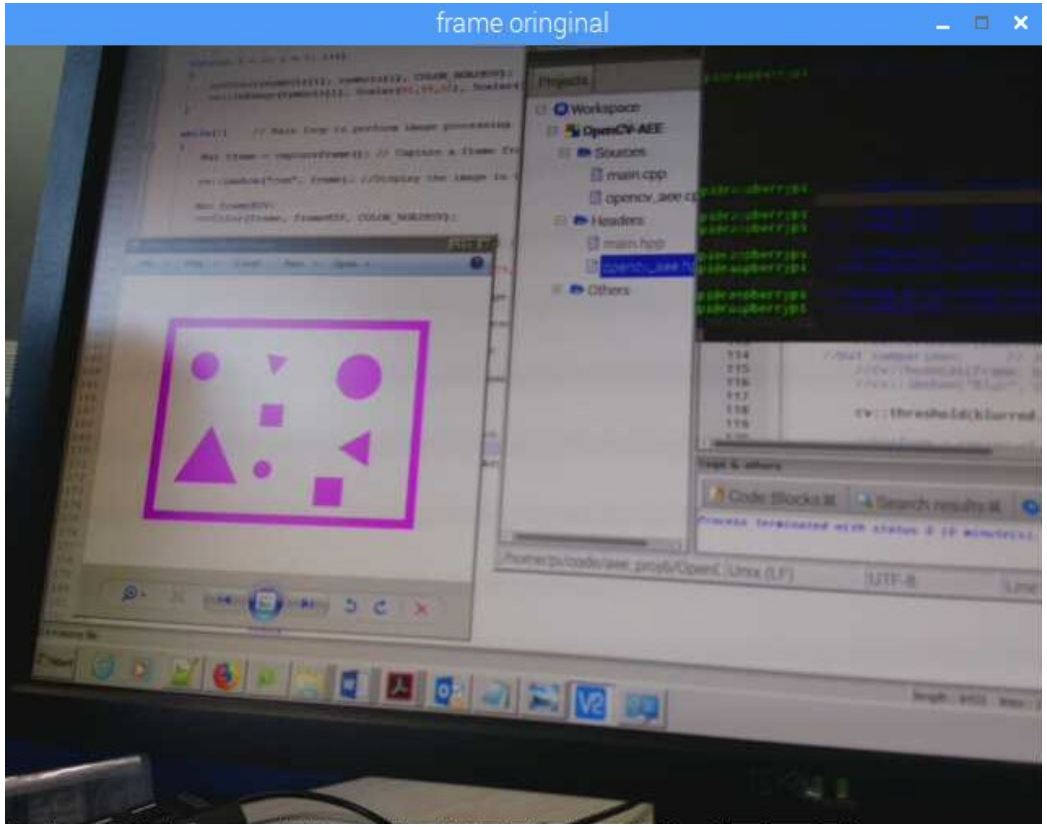


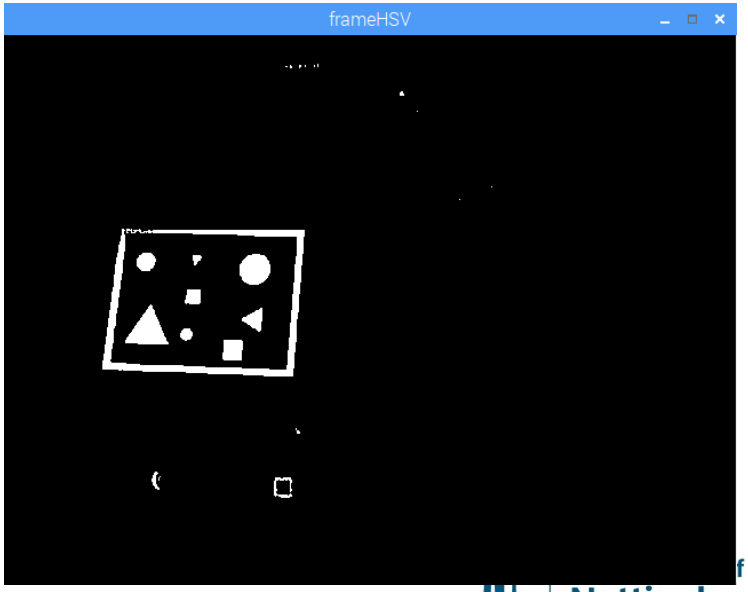
Let’s go through an example of template matching very quickly

EEEE1039:

OpenCV – Convert color

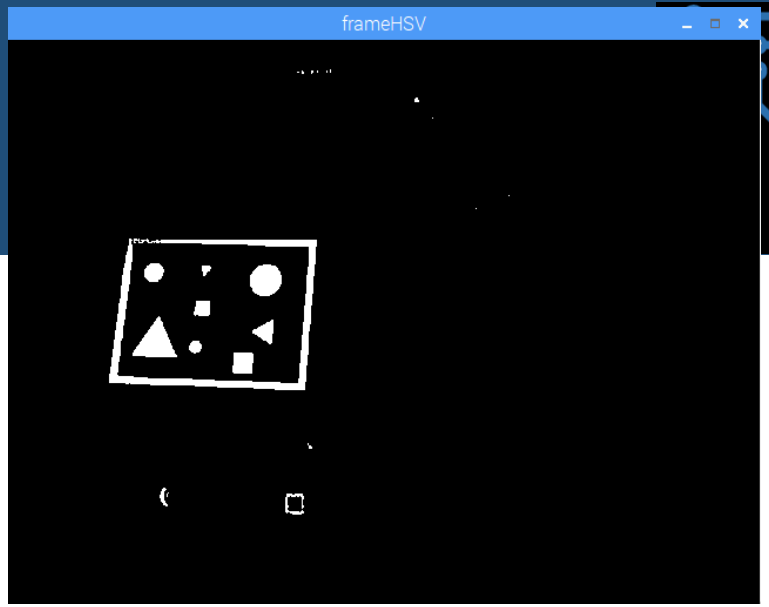


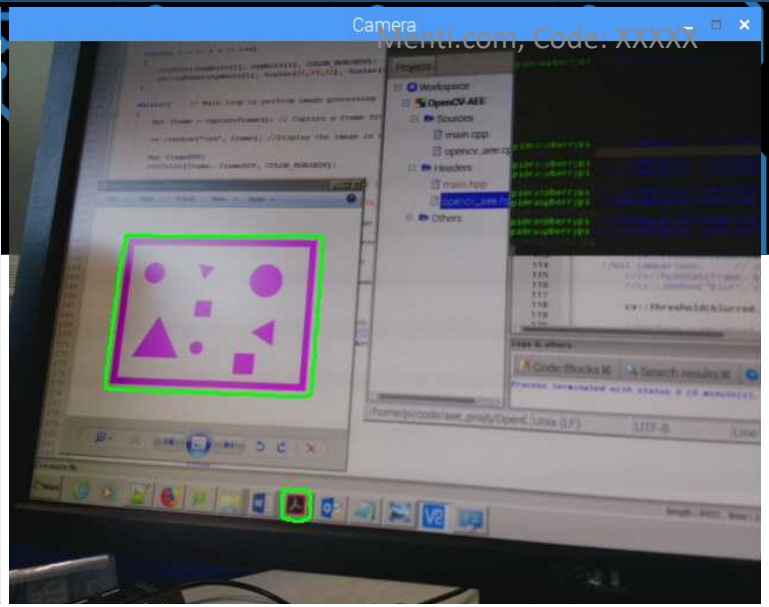




Due to the perspective of the camera
Rectangle -> Parallelogram

EEEE1039: OpenCV





Contour

- Find Contour

```
void cv::findContours ( InputOutputArray     image,
                        OutputArrayOfArrays  contours,
                        OutputArray          hierarchy,
                        int                  mode,
                        int                  method,
                        Point                offset = Point()
                      )
```

Image source is an 8-bit single-channel binary image.

Each detected contour is stored as a vector of points.

Information about the image topology

RETR_TREE: retrieve all contours and their hierarchy

CHAIN_APPROX_SIMPLE: end points only

```
std::vector< std::vector<cv::Point> > contours;
std::vector<Vec4i> hierarchy;
cv::findContours(dilated, contours, hierarchy, RETR_TREE, CHAIN_APPROX_SIMPLE,…
                 Point(0, 0));
```

https://docs.opencv.org/3.1.0/d3/dc0/group__imgproc__shape.html#ga17ed9f5d79ae97bd4c7cf18403e1689a

https://docs.opencv.org/3.1.0/d3/dc0/group__imgproc__shape.html#ga4303f45752694956374734a03c54d5ff

EEEE1039:

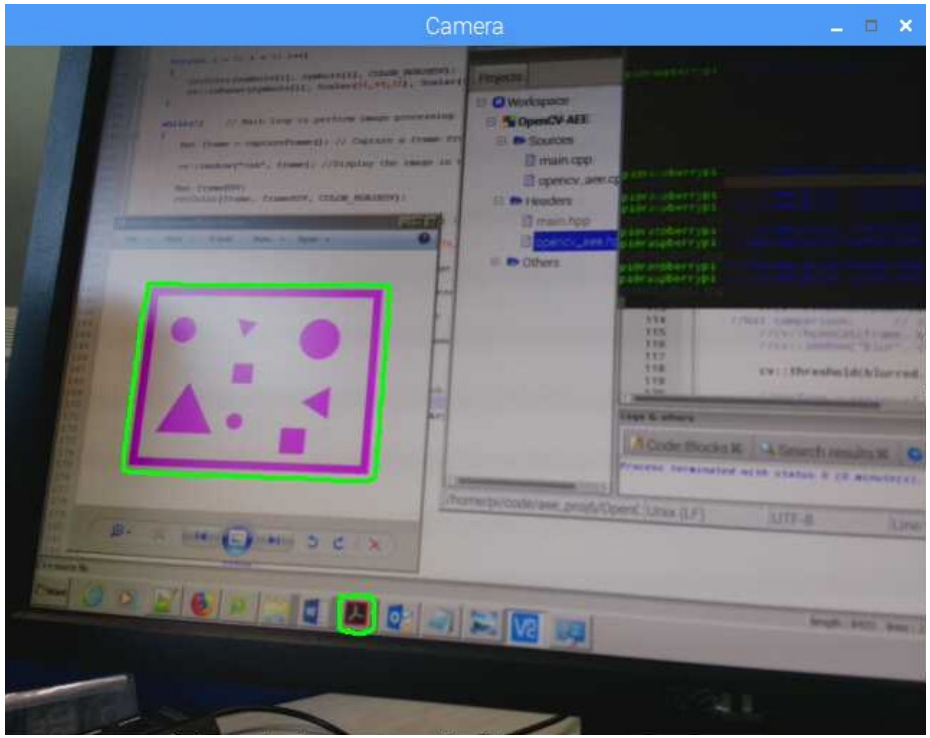
OpenCV – Contour

Contour

- Area of a contour
  - Useful in finding the largest contour in the image
    - The frame of the template

```
double cv::contourArea ( InputArray  contour,
                         bool        oriented = false
                       )
```

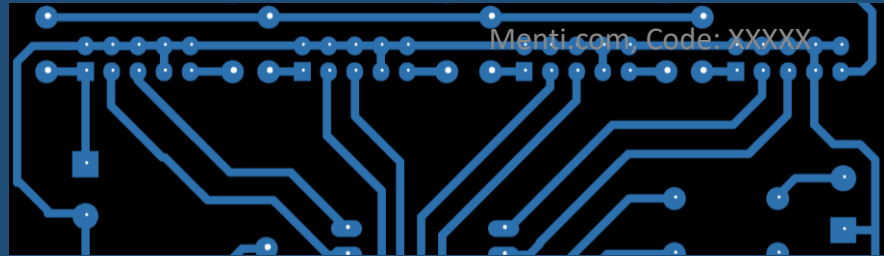
- Centre of a contour
  - Useful in finding the relative position in the image
    - Move the camera
    - Crop the image
    - Perspective transform




```
Point regionCentre = findContourCentre(contours[i]);
printf("Contour centre: x = %dpx, y = %dpx\n", regionCentre.x, regionCentre.y);
```

https://docs.opencv.org/3.1.0/d3/dc0/group__imgproc__shape.html#ga2c759ed9f497d4a618048a2f56dc97f1

EEEE1039:
OpenCV – Contour



Contour

- Contour Approximation
  - Approximates a polygonal curve(s) with the specified precision.

```
void cv::approxPolyDP ( InputArray   curve,
                        OutputArray  approxCurve,
                        double       epsilon,
                        bool         closed
                      )
```

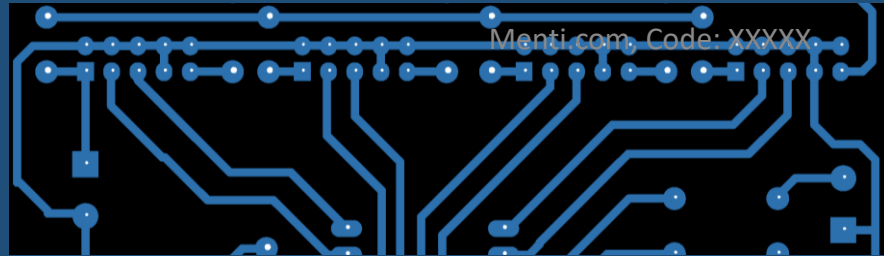
Approximated contours in green lines with bigger epsilon and smaller epsilon

- We need to approximate the symbol frame contour to a Parallelogram to do the perspective transform

Find more info here!

https://docs.opencv.org/3.1.0/dd/d49/tutorial_py_contour_features.html

EEEE1039:



Template Matching

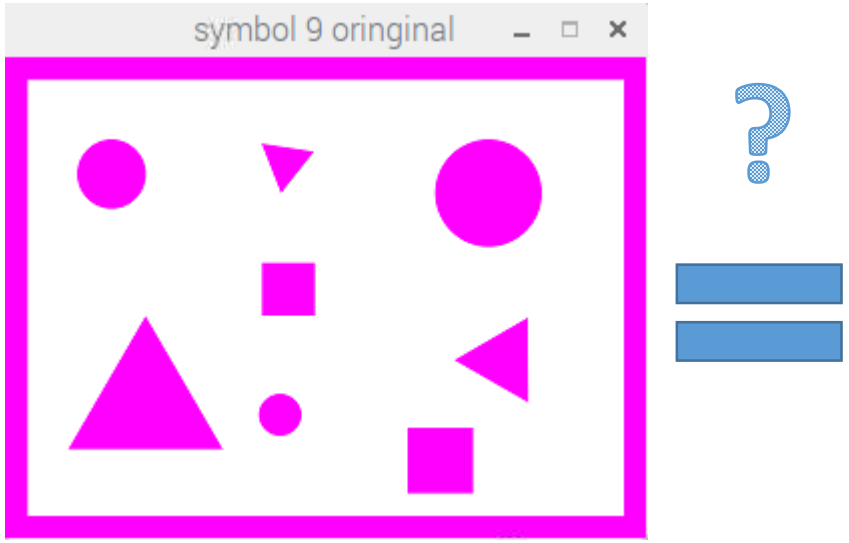
EEEE1039:

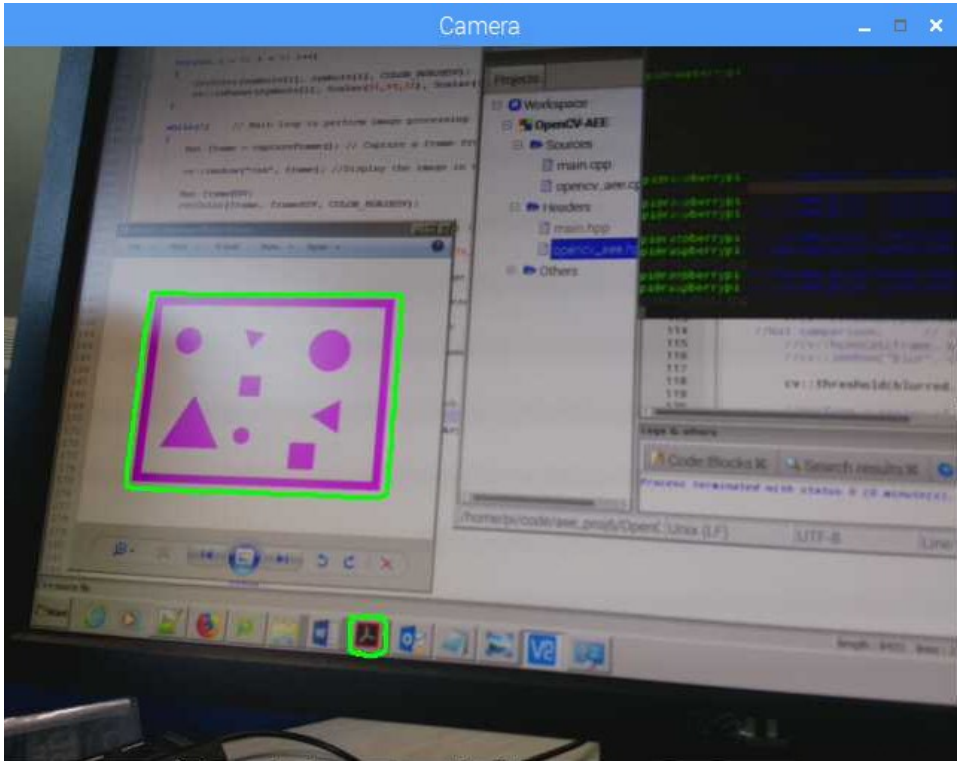
OpenCV – Template Matching

Menti.com, Code: XXXXX

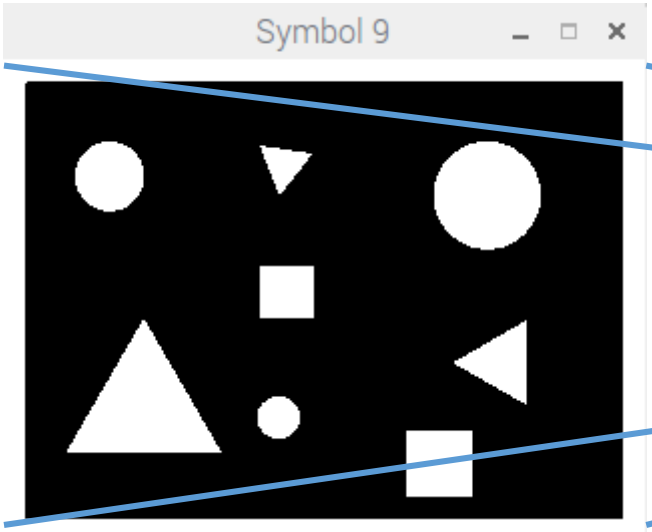
Template matching

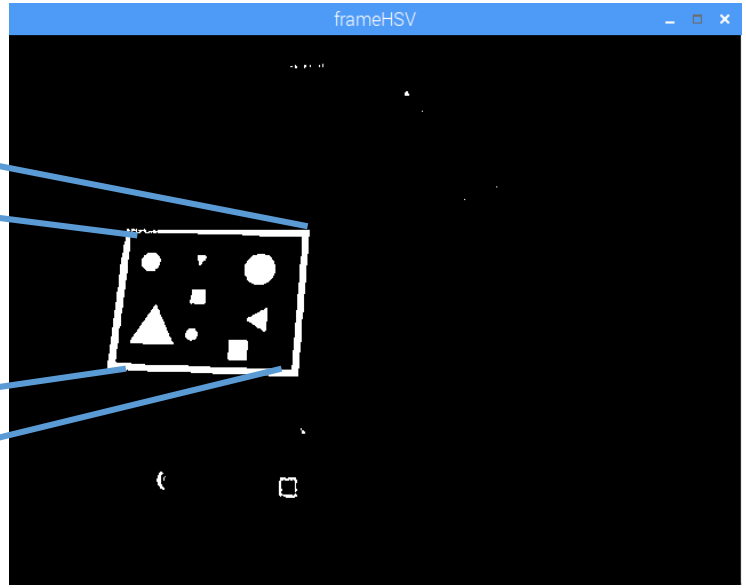
after identify a region of interest (ROI)



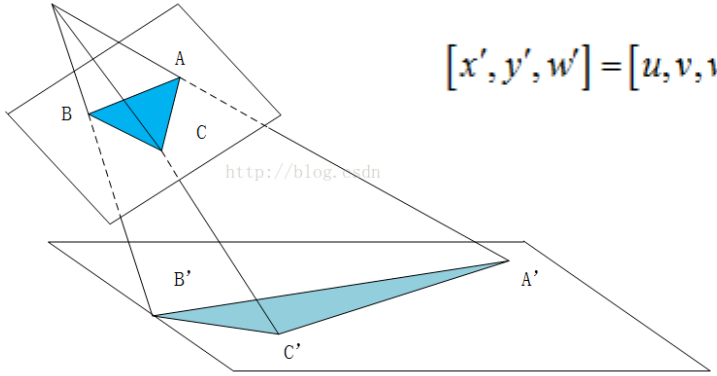


EEEE1039:
OpenCV – Template Matching





Perspective transfer



$$[x', y', w'] = [u, v, w] \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

To get the transformation matrix

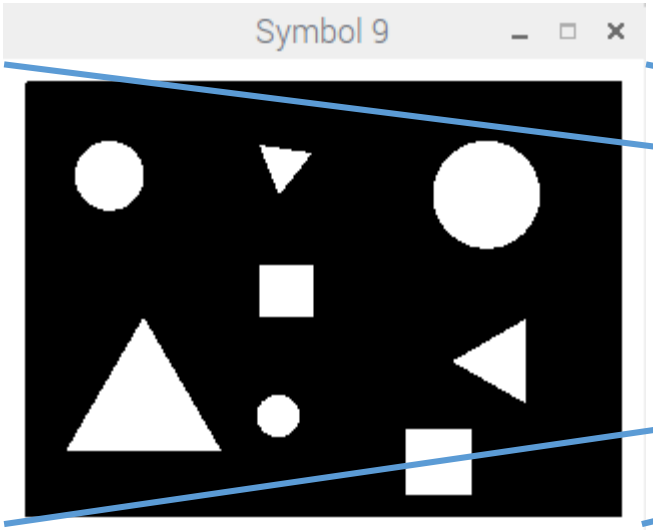
```
Mat cv::getPerspectiveTransform ( InputArray src,
                                  InputArray dst
                                )
```

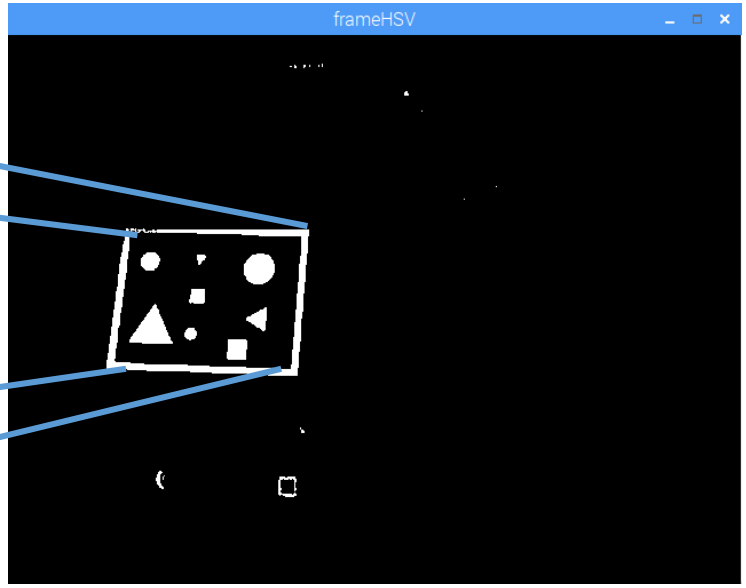
https://docs.opencv.org/3.1.0/da/d54/group__imgproc__transform.html#ga8c1ae0e3589a9d77fffc962c49b22043

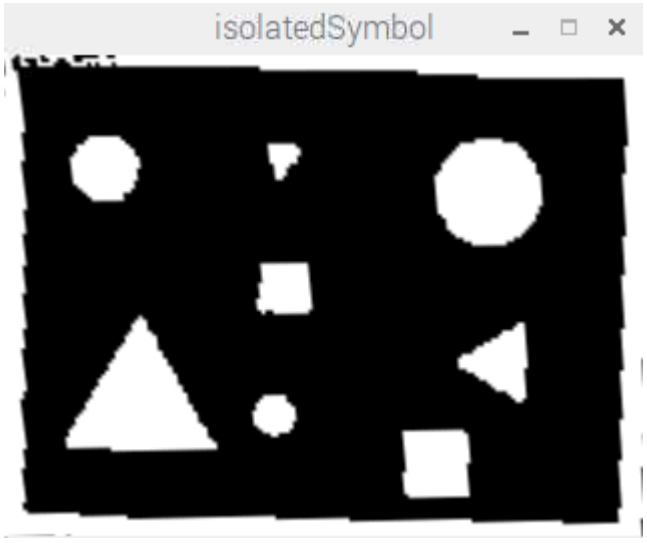
EEEE1039:

OpenCV – Template Matching

Perspective transfer



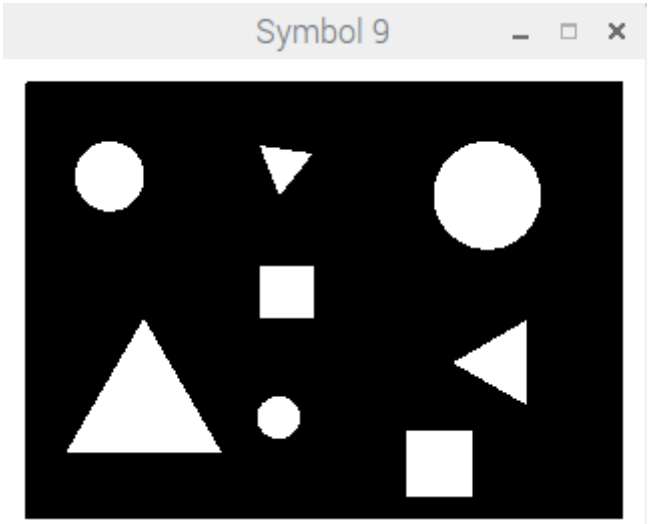


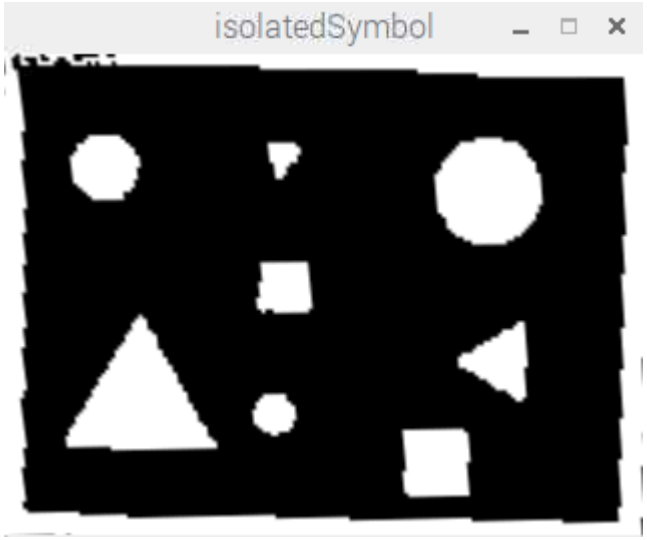



```
void cv::perspectiveTransform ( InputArray  src,
                                OutputArray dst,
                                InputArray  m
                              )
```

Or use the **transformPerspective()** function given in the AEE library

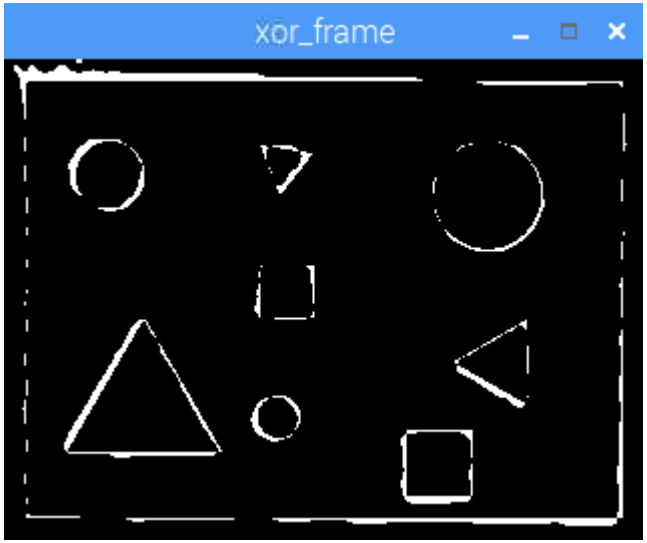
EEEE1039:
OpenCV – Template Matching





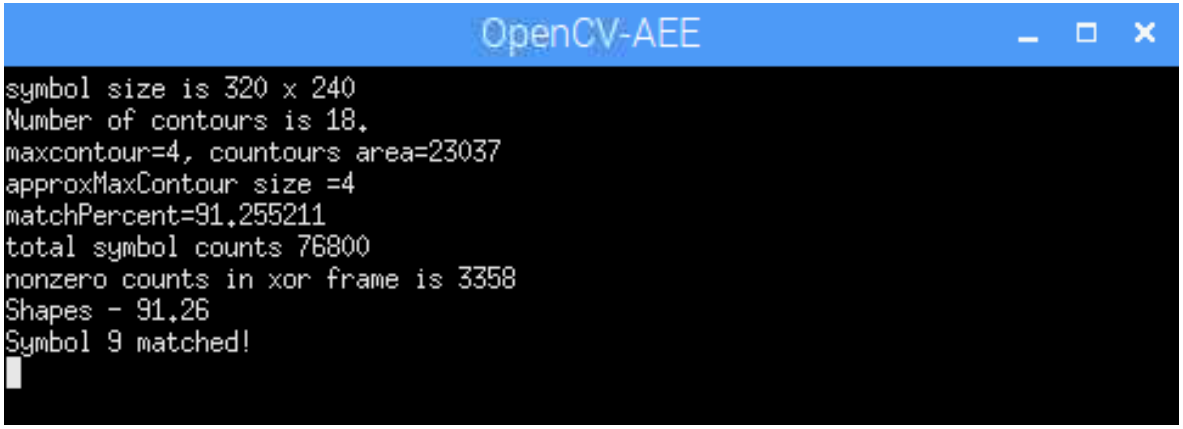
XOR

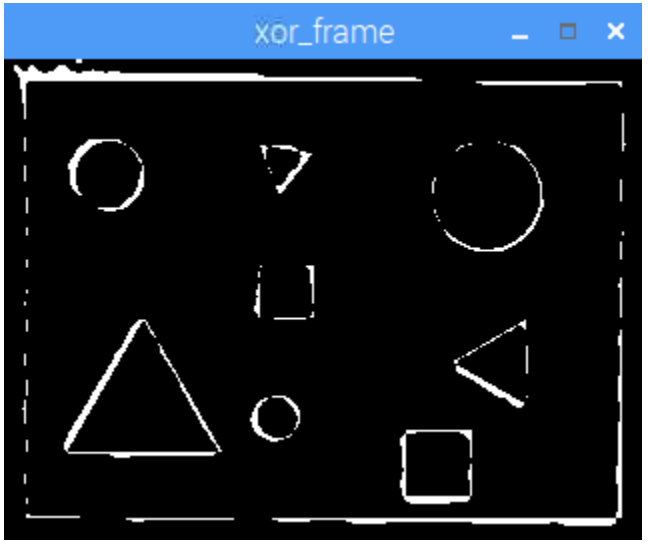
Similarity



- XOR the isolated image with symbol
- Non-Zero Pixels Indicate the difference

EEEE1039:
OpenCV – Template Matching



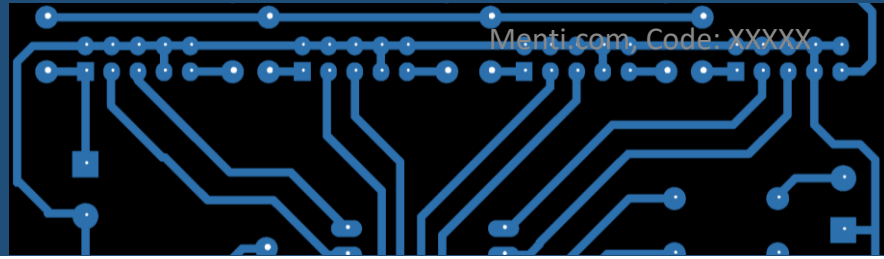


- Count Non-Zero Pixels in the XOR-ed frame
- Calculate the match rate

```
int cv::countNonZero ( InputArray  src )
```

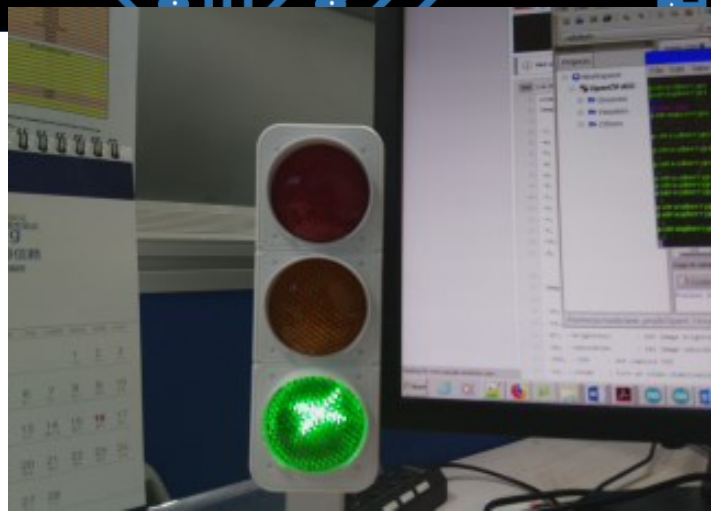
https://docs.opencv.org/3.1.0/d2/de8/group__core__array.html#gaa4b89393263bb4d604e0fe5986723914

EEEE1039:



Take *traffic light* as an example…

EEEE1039:
OpenCV – Traffic Light



One way of doing it

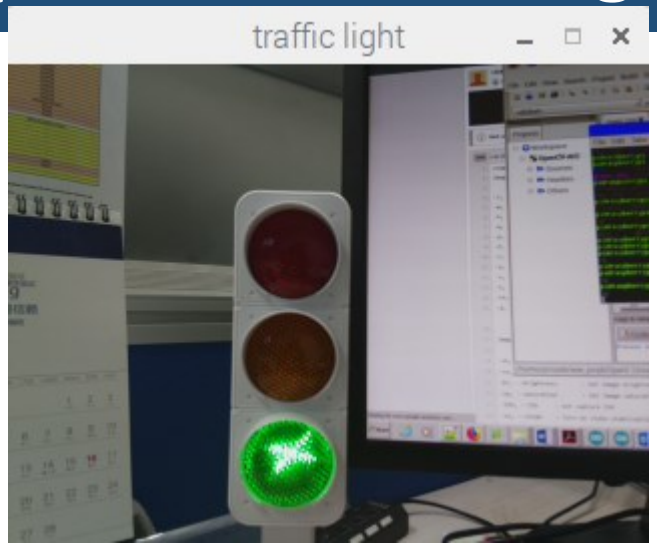
- Capture an image
- Generate the contour of the brightest spot
- Find the coordinate of the contour centre
- Compare RGB color of the contour centre pixel to determine which light is on

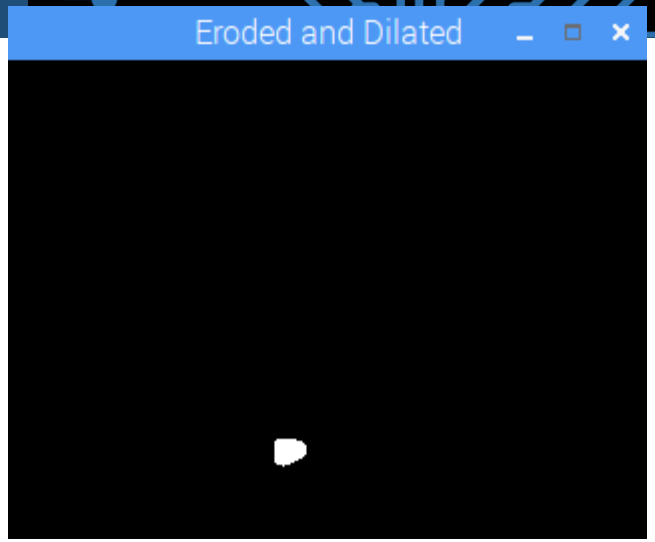
- Of course you are highly recommended to develop your own algorithm…

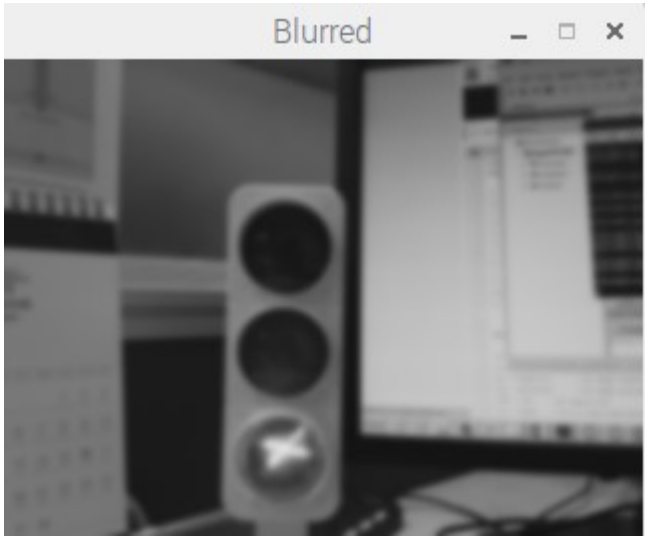
EEEE1039:

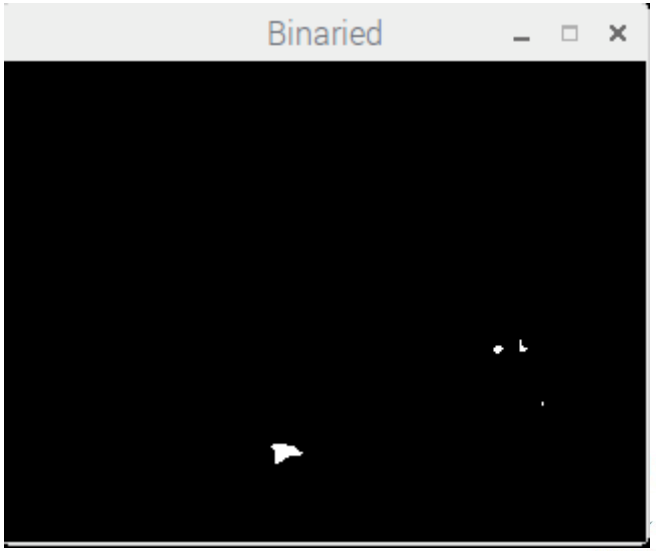
OpenCV – Traffic Light

Menti.com Code: XXXXX

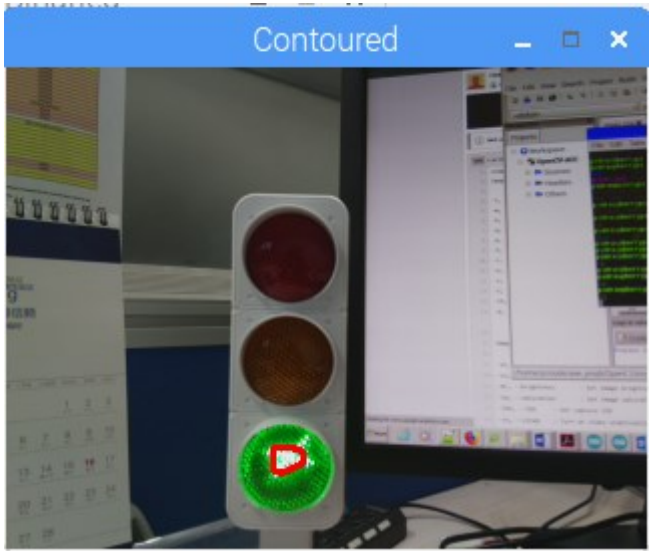


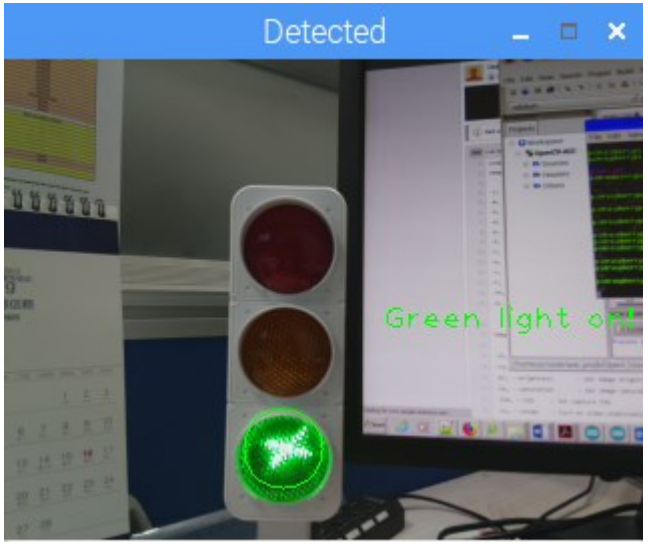






EEEE1039:
OpenCV – Traffic Light





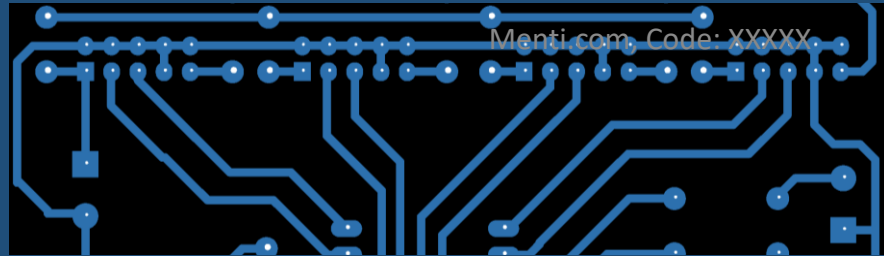
OpenCV-AEE

```
countours size is 1
Max countours idex=0, area=155
Contour centre: x = 139px, y = 194px
Contour centre color is 211, 226, 212
```

Contour centre RGB color (211,226,212)

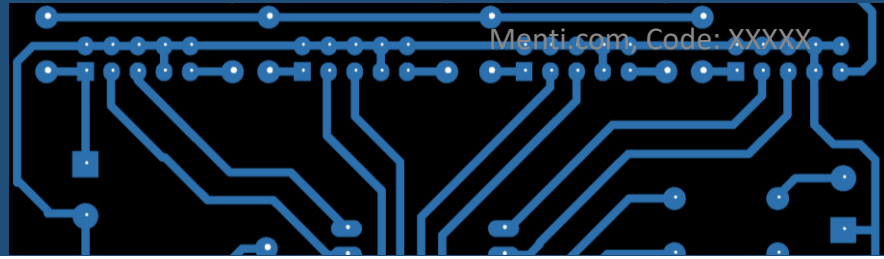
-> Green light on!

EEEE1039:

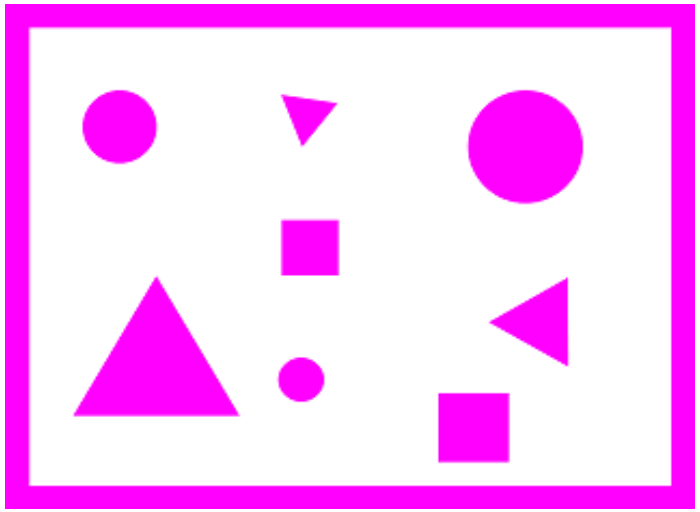


Take *counting shapes* as an example…

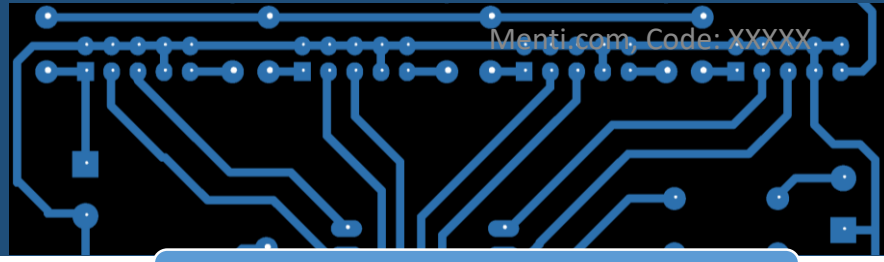
EEEE1039:
OpenCV – Counting shapes



- Take a sample photo to work on
- Identify key features of the image:
  - Colours
  - Borders
  - Corners
- Isolate these features
- Process the resulting images

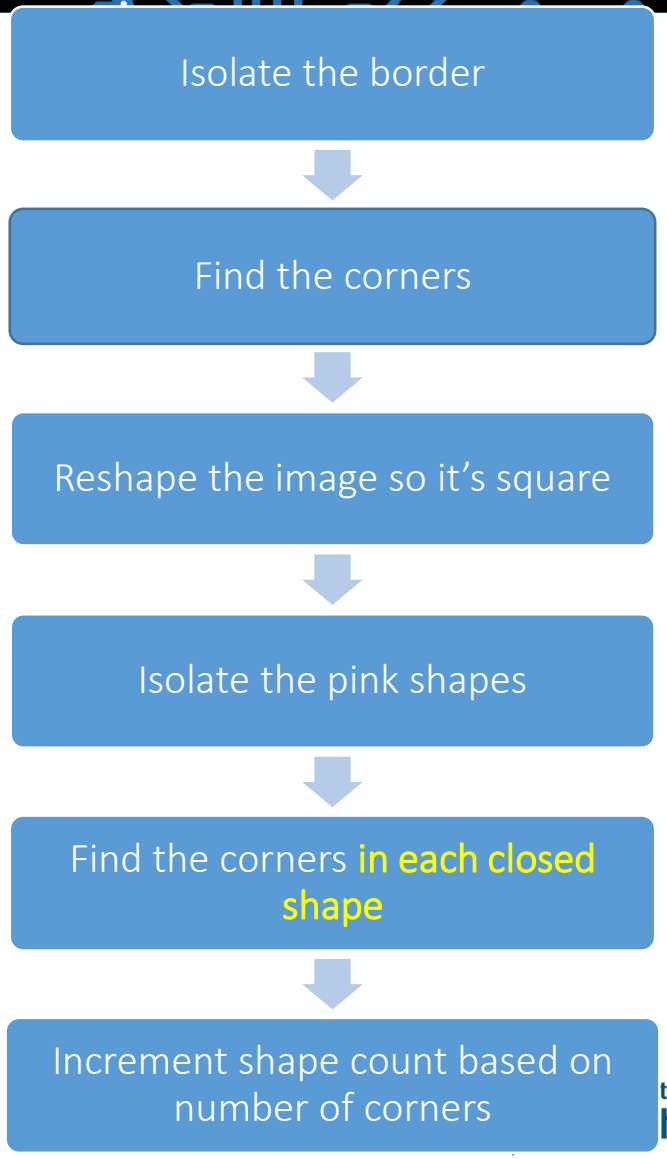


EEEE1039:
OpenCV – Counting shapes



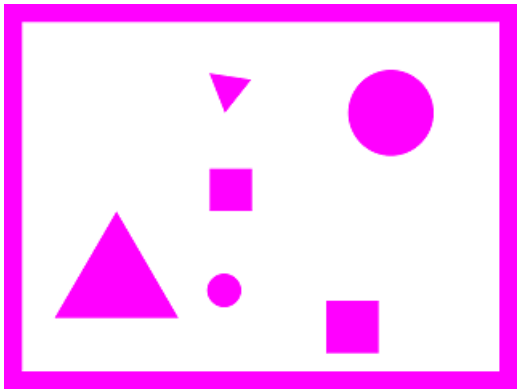
Good practise

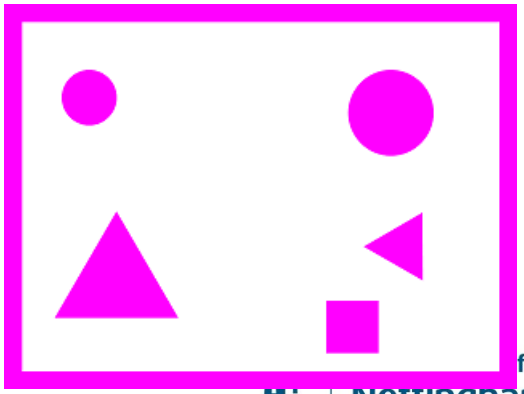
- Start with a flowchart
  - Include how the image should change at each stage
- When testing, display the image at each stage
  - Does it look correct?
  - Which bits are highlighted?
  - Does the lighting condition affect the result?
  - Etc…



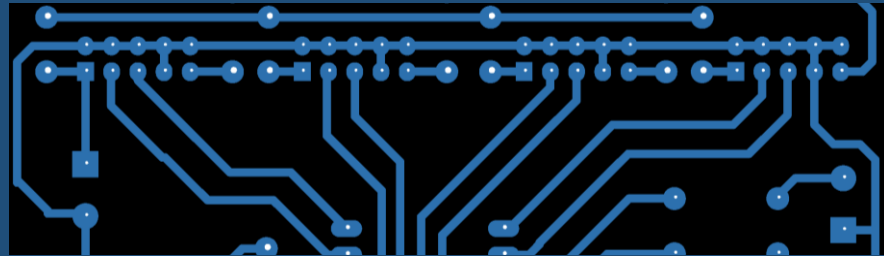
EEEE1039:

- Only 3 templates for “Counting Shape”.
- Simplify the algorithm through an easier way.



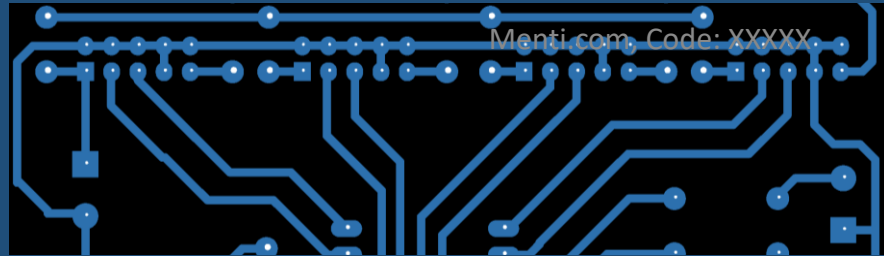


EEEE1039:
Content list



- Project 6 & 7 Introduction
  - General
  - Components and System
  - OpenCV
- **Assessments**
- Component recycle

EEEE1039:
Assessment List



Available on Moodle

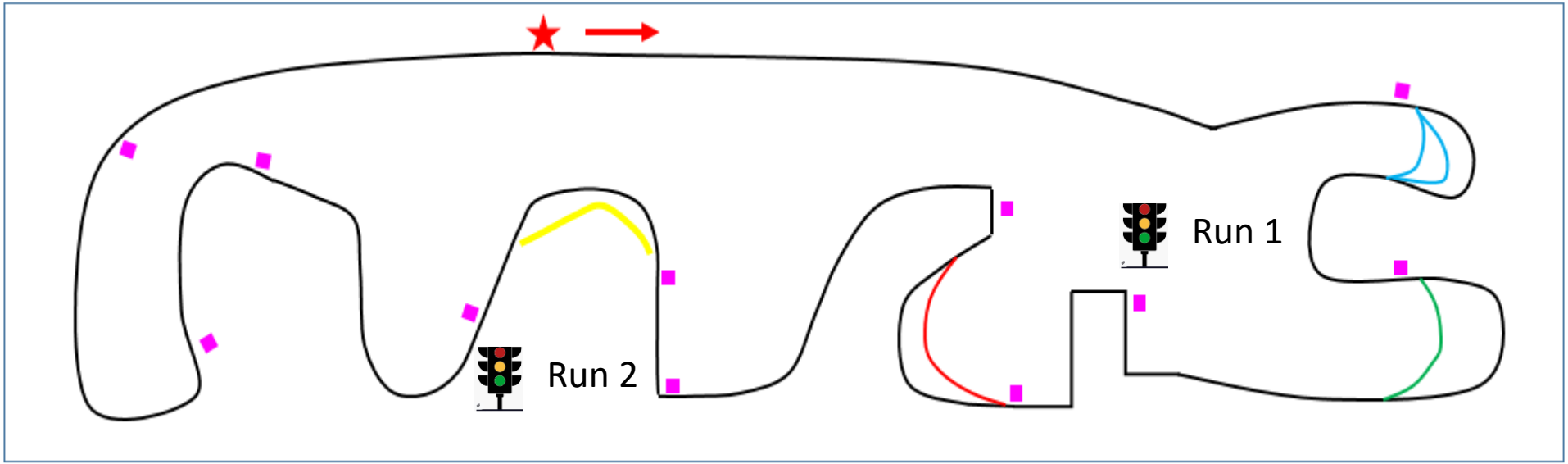
Assessment	Description	Semester	Weighting (%)
Demo 5	Demo of Session 5	Spring	2
Coursework 1	Coursework regarding OpenCV	Spring	5
Showcase	EEE & PDM showcase, demo, Q&A	Spring	5
Report 3	Report on Session 5 to 7 + PDM collaboration	Spring	15
Management 2	Planning 5-7, Logbook	Spring	5
CPD 2	Report + Reflection	Spring	5
Viva + Q&A	Viva voce + Q&A	Autumn + Spring	10
Finale	Final demonstration	Spring	10

EEEE1039:
Assessment – Final demo

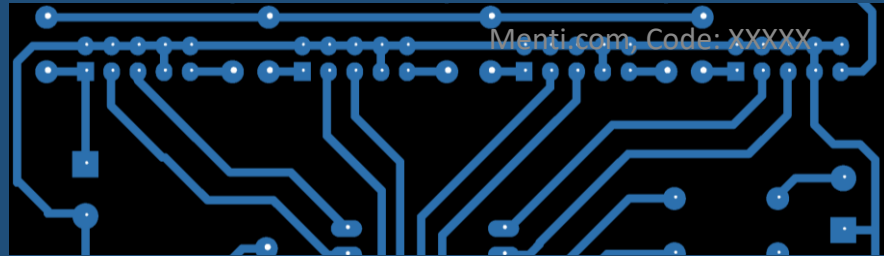
All tasks need to be completed **in two runs successively**, to be considered as task completed.

Finish @ 6pm sharp on Friday in Project Week 7. No time extension!

The CodeBlocks code for the final demonstration should be submitted before **8pm on Friday in Project Week 7**, i.e. the demo day.

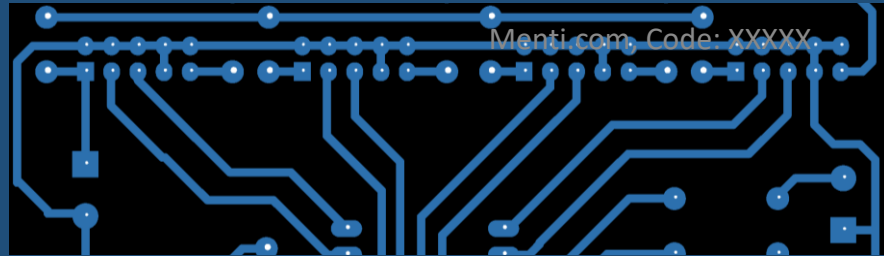


EEEE1039:
Assessment – Final demo



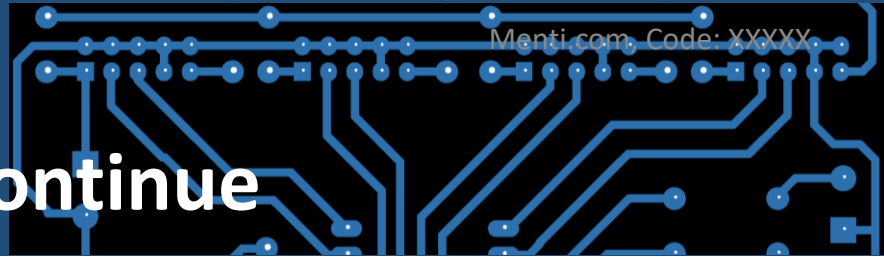
No.	Task	Mark	Description
1	Line following	4	Run the track twice successively
2	Count shapes	1	Count the number of each shape, and display on LCD
3	Short cut – blue	0.5	Run between split path
4	Short cut – green	0.5	Run with higher speed
5	Short cut – red	0.5	Run with lower speed
6	Short cut - yellow	0.5	Run with normal speed
7	Play music	1	Play a music or any audio
8	Alarm flash	1	Flash the red and blue LED alternatively
9	Approach and stop	1	Approach to the template then stop at 5cm distance
10	Kick football	1	**Bonus**! Kick the football to gate
11	Traffic light	1	**Bonus**! Stop for red light, and wait until green light shows
12	Multiple tasks	1.5	**Bonus**! Successfully complete task 1-9

EEEE1039:
Assessment – Final demo



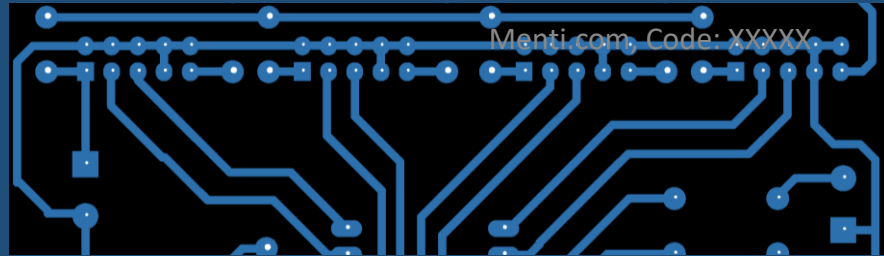
- Enjoy the project & challenges and **be nice to your teammates!!!!**
- **Minimum speed limit: >= 20%**
- **LCD displays recognized task in real time** (only for template matching, not for line following)
- Shortcut – if your vehicle can recognize the shortcut color, can just go ahead and no need to lift up the camera for symbol recognition
- Can request for demo anytime during project week 6 & 7 for either task(s), as long as your group is ready
- Can demo either task(s) you want
- Can demo as many times as you want until success
- **Cannot** remove or cover any pink square, short cuts or templates for demo

EEEE1039:
Assessment – Final demo - Continue



- Only components provided by this module are allowed to be used for demonstrations.
- LED lighting for demonstration is allowed. Again, LED provided by this module is allowed. Other lighting component/device is not allowed.
- Power bank is not allowed for demonstration.
- No human interference is allowed during demonstration.
- Be patient! Marks are far less important than the soft skills you will develop during this process, as long as you keep trying.

EEEE1039:
Demo – Line Following



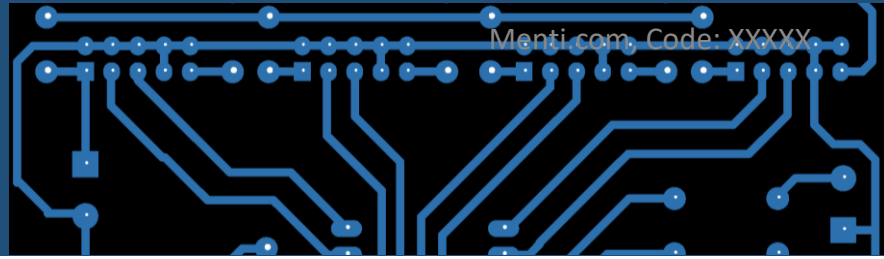
Requirements:

- Run the black track twice successively
- No need to recognize the pink squares

Tips:

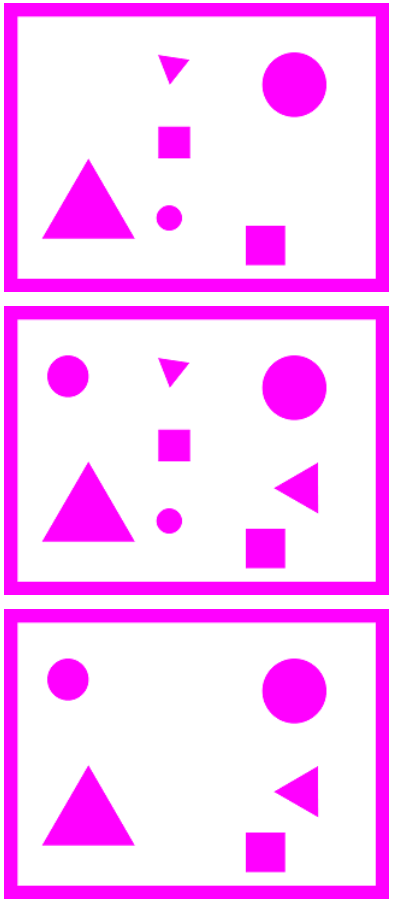
- Try to adjust the resolution for video capture of the Raspberry Pi camera, and observe how the resolution would affect the performance of the vehicle.
- Similar tip for template matching challenges. Try to adjust the camera resolution for template matching, observe how that would affect the accuracy in recognizing a template.

EEEE1039:
Demo – Count Shapes



Requirements:

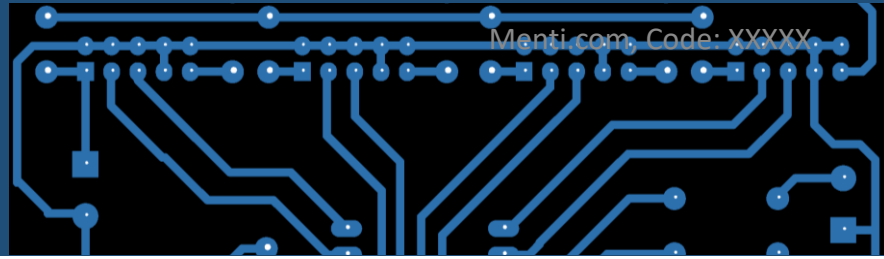
- Recognize the template, display on LCD **Count Shapes** for 5 seconds
- Recognize the numbers of circle, triangle and square, respectively.
- Display the number of each shape on LCD for 5 seconds, e.g. **C: 2 T: 2 S: 1**



Note:

- The outer-most pink rectangle which encloses all the shapes should **NOT** be included in shape counting.

EEEE1039:
Demo – Short Cuts

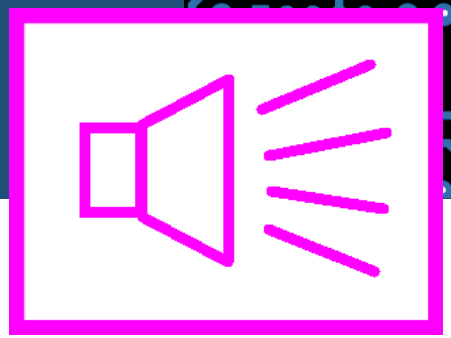


Requirements:

- Recognize the template, display on LCD "Short Cut - Color" for 5 seconds
- **Yellow**: normal speed, **red**: lower speed, **green**: higher speed, **blue**: through middle of two blue cuts
- **OK**: keep finding designated color(s) during line following routine, without lifting up the camera + recognizing templates, LCD shall display the recognized task while going through the short cuts.

Tips: After short cuts challenge, decision making to get back to the line following routine.

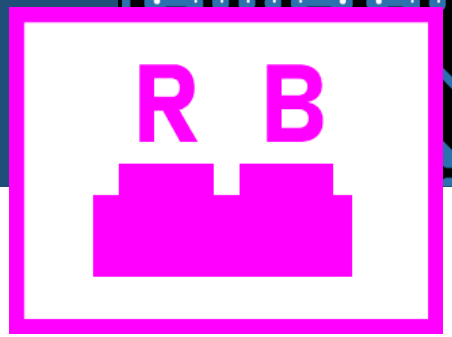
EEEE1039:
Demo – Play Music



Requirements:

- Recognize the template, display on LCD **Play Music** for 5 seconds.
- The vehicle keeps still.
- Play a piece of music or any audio which is **loud**, **clear** and **long** enough to be heard in the lab.
- After the music/audio play is done, the vehicle gets back to the line following routine.

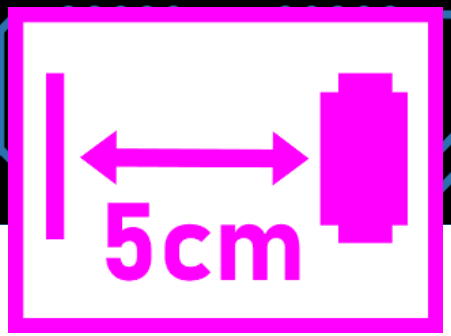
EEEE1039:
Demo – Alarm Flash



Requirements:

- Recognize the template, display on LCD **Alarm Flash** for 5 seconds.
- The vehicle keeps still.
- Raspberry Pi signals one **red** LED and one **blue** LED to blink alternatively for 10 seconds, while the blinking of each LED lasts for 1 second.
- Turn off both LEDs, then the vehicle gets back to the line following routine.

EEEE1039:
Demo – Approach and Stop



Requirements:

- Recognize the template, display on LCD **Approach and Stop** for 5 seconds.
- The vehicle runs forward towards the template placed in the front, then stops at a distance of **5±2cm** in front of the template.
- Display the distance measured by HC-SR04 with a precision of one decimal point on LCD for 5 seconds, e.g. **Distance: 5.2cm**
- The vehicle retreats back to the track, and continue with the line following routine.

EEEE1039:
Demo – Kick Football

BONUS

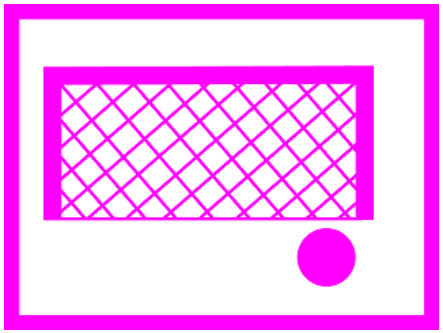
Menti.com Code: XXXXX

Requirements:

- Recognize the template, display on LCD **Kick Football** for 5 seconds.
- The vehicle should turn to the football, kick the football into the gate, then gets back to the line following routine.

Tips:

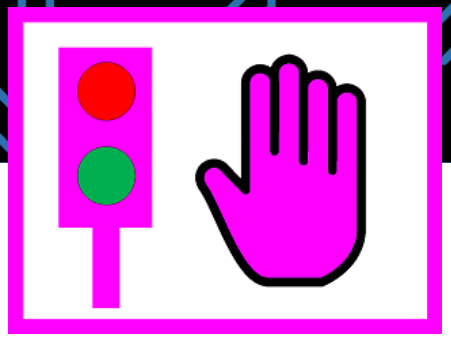
- Turn a fixed angle
- Recognize the football



EEEE1039:
Demo – Traffic Light

Menti.com Code: XXXXX

BONUS



- **Requirements**:
- Recognize the template, display on LCD for 5 seconds.
- The vehicle should keep still, and wait for the green light to be on, then can gets back to the line following routine.
- The time duration before the green light on is random.

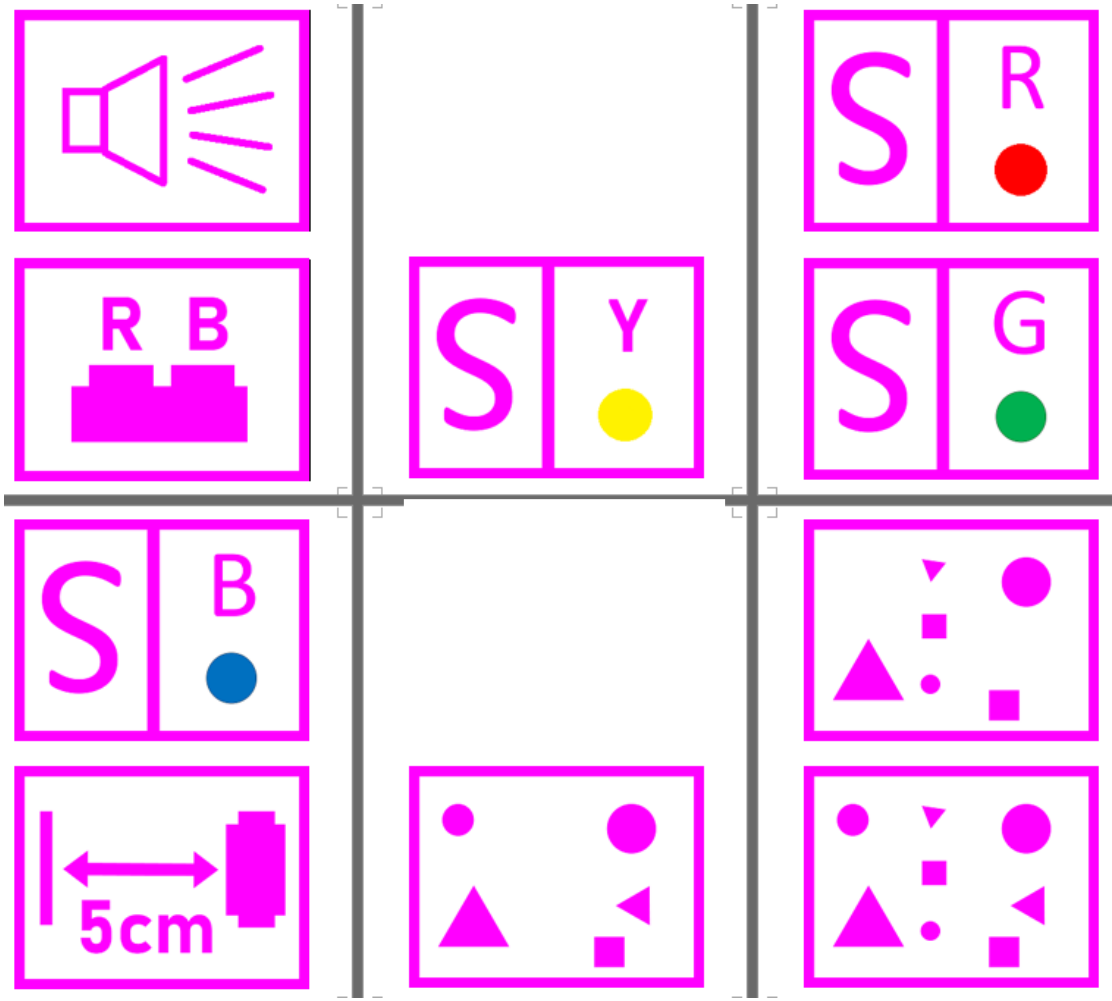
EEEE1039:
Demo – Multiple Tasks

BONUS

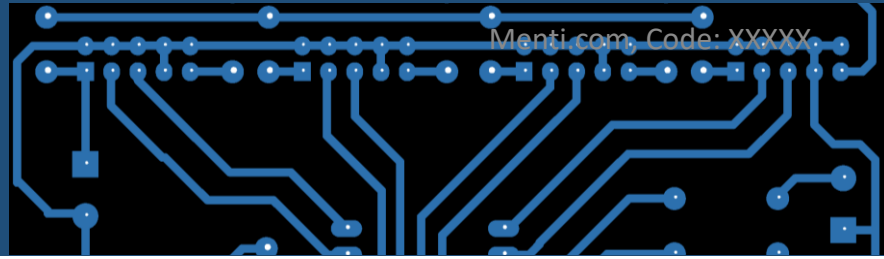
Menti.com Code: XXXXX

Requirements:

- In both successive runs, complete tasks on the right - requirements of each task are fulfilled
- Traffic light and football challenges are not included
- Short cuts: complete 4 colors

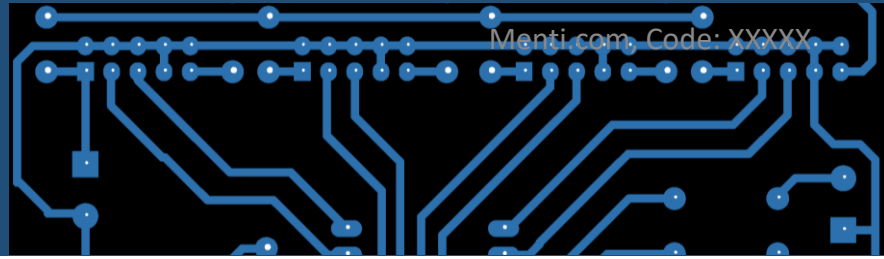


EEEE1039:
Assessment – Showcase



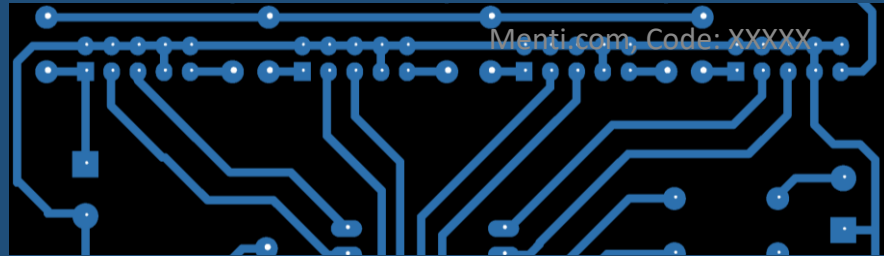
- Details are introduced in lecture 1 in spring semester.

EEEE1039:
Assessment – Lab report III



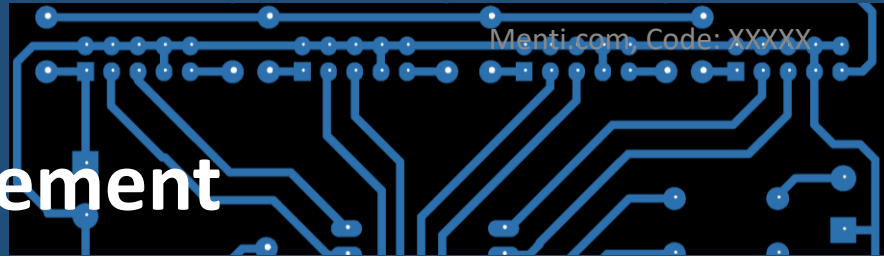
- Submission deadline: **3 pm, 3 April 2025**
- Individual report, <= 15 pages
- Content to be covered: project 5, 6, 7
- Codes will be submitted after the final demo, so **don't** put codes in the appendix, unless different and necessary
- Flow chart or pseudo code should be included in the main body
- A marking scheme for report No. 3 will be released on Moodle soon

EEEE1039:
Assessment – Log book



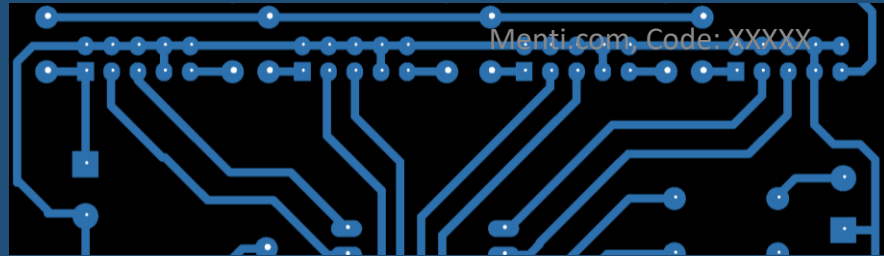
- Submission deadline: **3 pm, 11/25 March 2024**
- Upload the scanned copy to Moodle, hard copy is not accepted
- Will be assessed by your personal tutors
- Refer to the requirement in log book – “Keep a laboratory log book” on Moodle

EEEE1039:
Assessment – Project management



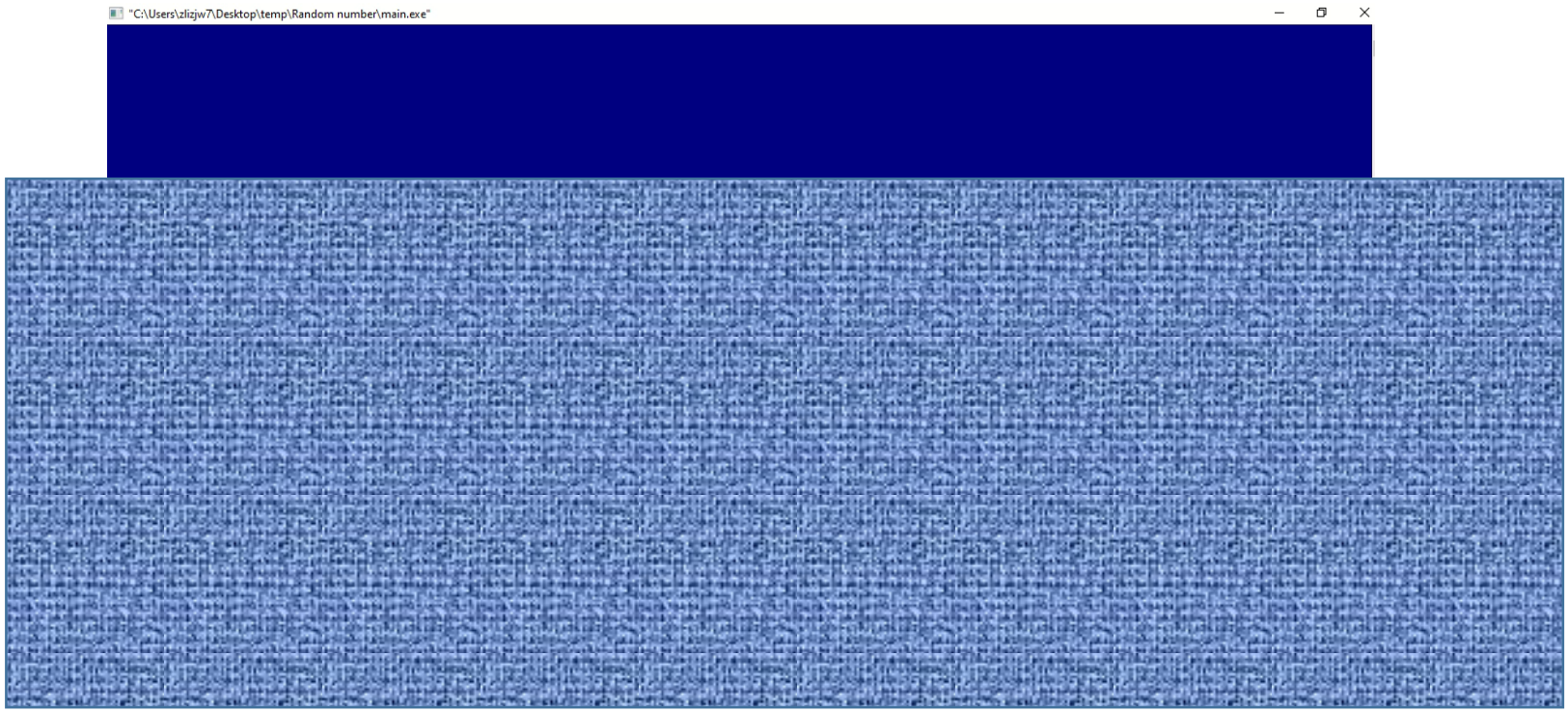
- Project planning (group) + CPD II (individual) should be submitted on time. Refer to “EEEE1039 Timetable 2024 Spring” on Moodle for deadlines
- Missing or delay in submitting project planning induces ZERO mark for management of that project week
- Planning must be approved by personal tutors to be considered as valid
- Planning should be detailed both in task assignments and person in charge
- Requirement on CPD is available on Moodle, same as autumn semester.

EEEE1039:
Assessment - Viva

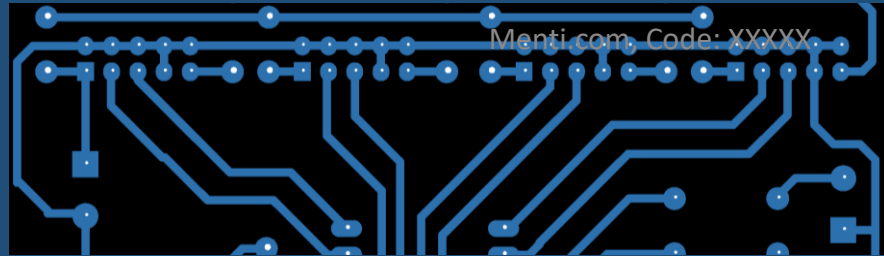


- Group viva, **11 April 2025**
- Location & viva schedule will be released one week before
- Content: **all 7 project weeks** in both autumn and spring semester
- Group presentation + individual Q&A + **ppt slides submission**
- Duration: 25 min = 15min pre + 10min Q&A
- Dress code: *Business Casual*

EEEE1039:
Viva Q&A

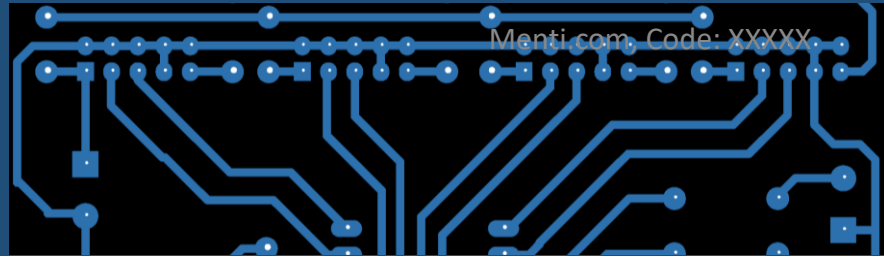


EEEE1039:
Content list

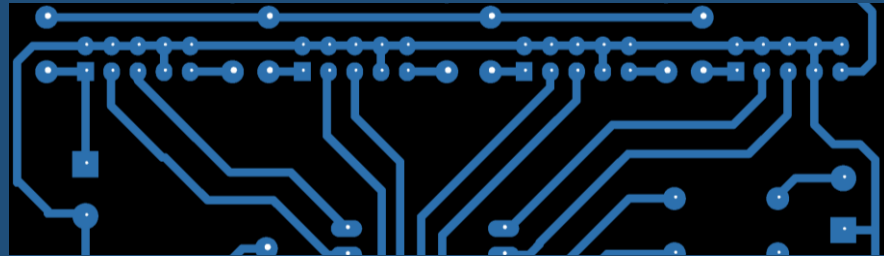


- Project 6 & 7 Introduction
  - General
  - Components and System
  - OpenCV
- Assessments
- **Component recycle**

EEEE1039:
Component Recycle



- The list of components to be recycled:
  - Autumn components: Already indicated in the components lists
  - Spring components: Available on Moodle
- 2 time slots scheduled on **17 April 2025**
  - 9:30 am – 12 noon
  - 1:30 pm – 4:30 pm
  - Any special circumstance, email me
- Venue: PMB207
- **Keep the vehicle as what it looks like on Friday in Project Week 7**
- Report to Ms. Gigi Lu in PMB207 if any requested recycle component is missing



Thanks!