

# Lesson 1 Visual Lab

Subject: Computer Science/Design and Technology Level of Difficulty: Intermediate **Topic:** Visual Recognition **Efforts:** 60 minutes



#### Learning Goals

- What is Smart Camera? How does it work?
- What is PixyMon? How does it work with Smart Camera?
- How to use labels to control mBot's movement?

### 🗄 Learning Content

#### Smart Camera





Smart Camera (see Figure 1) is a vision sensor that can extend mBot's ability to "see" and enable the robot to recognize things. It mainly has three features: **color block detection**, **line detection**, and **barcode label detection**.

**Color block detection**: Smart Camera can learn single-colored or bright-colored objects, and report more accurate results when detecting objects in a brighter color.

**Line detection**: When we switch Smart Camera to this mode, the module can detect lines, measure the number of branches of an intersection, and report information, including coordinates, directions, and the number of branches.

**Barcode label detection**: Smart Camera can also recognize the barcode labels included in the Smart Camera Kit without learning.

#### PixyMon

#### 1. Features



PixyMon is the tool that allows you to teach and debug Smart Camera. In PixyMon, you can view the images captured and returned by Smart Camera, the color blocks it learns, and the lines detected. Also, you can adjust parameters for higher detection accuracy.

#### 2. Connection

To input the images to PixyMon, you need to connect the module to your computer via a micro USB cable. When the connection is done, the images appear on the PixyMon window.



Figure 3 Connect Smart Camera to your computer

#### 3. Operating modes

PixyMon allows you to switch Smart Camera between operating modes. The two options marked in red in the figure below are responsible for color blocks detection and label detection respectively. Both line detection and label detection are under the same mode, **line\_tracking**.



Figure 4 PixyMon window

Note: Color block detection and line detection can't run at the same time. However, barcode label detection and line detection are under the same mode, so they can run at the same time.



#### Label detection

The Smart Camera Kit includes 15 labels numbered from 1 to 15. Target Smart Camera at a label and see its number on PixyMon.



Figure 5 Labels for Smart Camera to detect

Each label has its own number. The numbers and conditional statements are useful

elements in a program when you need to control mBot's vision.

-	E	e s	ma	irt ca	am	era	р	ort1	•	detected label (1) forward ▼ ?
				100			2		2	· · · · · · · · · · · · · · · · · · ·
										🗸 (1) forward
										(2) backward
										(3) turn left
										(4) turn right
										(5) stop
										(6) plus sign
										(7) minus sign
										(8) question mark
										(9) red heart
										. (10) A
										(11) B
										(12) C

Note: The text after the numbers in this block explains what each label can do.

To use the block above, you need to switch Smart Camera to the right mode—

Line/label Tracking. The block below is the one you need:



#### Hands-on

#### 1. Task

Use labels to control the movement of your mBot.

#### 2. Install Smart Camera

The picture below shows how to connect Smart Camera to mBot:



Note: Smart Camera needs its own power source, so you need to connect an extra Power module to it.

Your mBot should be like this when you finish building it:





#### 3. Algorithm Design

Use a flowchart to describe the logic of your program. Draw it in the space below:





# ⊞ Quiz

1. Which mode in PixyMon is for barcode label detection?

A. color\_connected\_components B. line\_tracking

2. Can we use the barcode labels to control mBot without executing any program?A. YesB. No

# 🛱 Extension

The lead of the Visual Lab says that they're studying how to apply Smart Camera devices to cashier-less stores. Can you design a self-checkout device by using the label detection feature of Smart Camera?



# Lesson 2 Object Recognition I

Subject: Computer Science/Design and Technology Level of Difficulty: Intermediate **Topic:** Visual Recognition **Efforts:** 60 minutes



#### Learning Goals

- How does mBot with Smart Camera recognize a colored object?
- How to teach mBot color block detection skills?

## 🗄 🛛 Learning Content

#### **Color Block Detection**

#### 1. Color block learning

Under the color block detection mode, Smart Camera can learn the color of an object. Smart Camera can learn up to seven different color blocks. If you want it to learn more, then it has to "forget" the blocks it learned to memorize the new ones.

#### Follow the steps below to teach Smart Camera a color block on PixyMon:

- Connect Smart Camera to your computer via a USB cable;
- Open PixyMon;
- Go to **Program** in the menu bar, and select **color\_connected\_components**;
- Place the object you want to teach in the field of view of Smart Camera (see 1) of Figure 1);
- Go to Action > Set signature 1. When you do this, the PixyMon window freezes (see

   of Figure 1);
- Press the left mouse button and drag the mouse to select the color you want to identify (see ③ of Figure 1);

PixyMon returns to dynamic display when Smart Camera finishes learning. The detected color block will be in a rectangular frame with its number (see ④ of Figure 1).



Figure 1 How to teach Smart Camera a color block

#### 2. Region detection

An image returned by Smart Camera can be divided into five regions. Under the color block detection mode, Smart Camera can detect which region a color block is in. The regions are as shown below:

To make Smart Camera identify which region a color block is in, you need this coding block:





#### Hands-on

#### 1. Task

Make mBot follow a ball automatically.

**Question 1:** How to control mBot's movement through region detection?

**Question 2:** To act out effective detection, where should you install Smart Camera?

#### 2. Algorithm Design

Use a flowchart to describe the logic of your program. Draw it in the space below:



#### Note: Angle and detection

The angle and height at which you install Smart Camera will affect the detection result. You would do well to adjust the installation position depending on the object you want to track.

Below are two situations where the positions of the ball are the same but the installation angles of Smart Camera are different. In the first situation, Smart Camera can "see" the ball within the central scope of the camera frame. In the second situation, the module can't "see" the ball.

We can see that in the second situation, Smart Camera is parallel to the ground and this restricts its field of view, so it can't "see" the ball unless the ball enters its scope.







### Learn More: Teaching Smart Camera without PixyMon

If you want to detect a ball of another color, do you need to return to PixyMon to teach Smart Camera again? Can you find a better and faster way to teach the module?

These are the stages of teaching Smart Camera the first object offline:

- Press and hold the **Learn** button on the right side of Smart Camera;
- The indicator light at the back turns on white first, and then starts a series of flashes, from red, orange, yellow, green, cyan, blue, and purple. To teach Smart Camera the first color block, release the button when the indicator turns red.

Red	Orange	Yellow	Green	Cyan	Blue	Purple
Color						
block 1	block 2	block 3	block 4	block 5	block 6	block 7

(**Note:** The seven colors are NOT related to the color of the object, and used only to refer to the color block number. For example, color block 1 can be a blue ball, even though color block 1 is displayed by red light, and color block 6 can be an orange object while color block 6 is indicated by blue light.)

- Hold the object in front of Smart Camera. When the indicator matches the color of the object, press the Learn button again to confirm Smart Camera's learning outcome.
- Smart Camera has learned the object.



# ⊞ Quiz

1. In the color block detection mode, how many colors at most can Smart Camera learn?

A.1 B.3 C.5 D.7

2. How many regions are the detection area divided into?

A.1 B.3 C.4 D.5

# 🛱 Extension

The experiment shows that mBot can track things. But that depends on whether the object is within the detection range. What if the object is out of the range? How to make mBot track that object?



# Lesson 3 Object Recognition II

Subject: Computer Science/Design and Technology Level of Difficulty: Intermediate **Topic:** Visual Recognition **Efforts:** 60 minutes



#### **Learning Goals**

- How to detect a color block in the coordinate system of Smart Camera frame?
- What is the relationship between the Kp value and mBot's movement?

## Learning Content

#### Color block detection and coordinates

An image returned by Smart Camera can be presented with a two-dimensional coordinate system (orthogonal coordinate system). The system helps increase the accuracy of detecting color blocks.

The values of the x-axis lie between 0 and 319, increasing from left to right. And the values on the y-axis increase from top to bottom, ranging from 0 to 239. The x-y coordinates of a color block can tell you its position on the image.



#### Figure 1 Range of coordinates

This coding block can help obtain the coordinates of the center of a color block. You can use them in your program.

🖞 smart camera	port1 🔻	detected	x-coordinate 🔻	of color	bloc	k 1	•	
		. √ x-c	oordinate					
		· y-c	oordinate					
		wic						
		nei	ight					
<b>PixyMon</b> e Program Action View	Help							
	neip							
n 🕹 🍕								
▲ <b>* *</b> ©				The center o	of the	dete	cted	color
	1			The center o	of the	dete	cted	color

**Note:** The coding block above can also measure the length and width of the rectangular frame around the color block.

Think about this question: If the object appears at the center of the image, what are supposed to be its coordinates?

The **Event-specific Smart Camera** category includes an encapsulated coding block that can calculate the distance between the current coordinates and the target coordinates. (If you're interested in the algorithms behind the block, go check the Algorithm Analysis part.)

#### Control the motor

If you want mBot to track and follow an object more smoothly and continuously, you need to set the Kp in motor differential speed. The Kp value is important in tuning mBot's steering sensitivity. It has a value range of 0 to 1. Generally, a higher Kp value and higher speed are necessary for making sharp turns. However, such setting causes stronger vibration when mBot turns. Use coding block below to set the Kp value. The Kp in motor differential speed calculation is encapsulated in this block.

🚆 smart camera port1 🔻 sets Kp in motor differential speed calculation to 0.3

#### Hands-on

#### 1. Task

You can calculate the distance between the coordinates of the color block and the target coordinates, which makes mBot follow a color. Then you can use the calculation result to control the rotational speed of the motor.

Question: What's the relationship between the Kp value and mBot's movement?

Adjust the parameters in the example program, and see what happens to mBot.

Find the parameters for the smoothest movement and take notes of your observation to fill in the table below.

Kp value	Coordinates	Smoothness and position
0.1	(160,170)	Not smooth or continuous; center
0.3	(160,170)	
0.4	(160, 170)	
0.5	(160, 170)	
( )	(160,170)	



#### 2. Algorithm Design

Use a flowchart to describe the logic of your program. Draw it in the space below:





# ⊞ Quiz

1. Higher speed and higher Kp value result in (	) steering sensitivity.
---	-------------------------

A. greater B. lower C. the same

2. If the target coordinates are set to (80, 170), and the object is detected not moving, then the color block is ( ) Smart Camera.

A. right ahead of B. to the left front of C. to the right front of D. below

# 🛱 Extension

Repeatedly uploading programs is time consuming during tests. Can you use buttons to adjust the speed and Kp value and program the LED Matrix to display the real-time speed and Kp value?



#### Learn More: Kp value and the motor motion

😬 smart camera port1 🔹 calculates motor differential speed (auto follows color block 1 🔹 to 🗴 🔹 axis (150) )

This block allows users to set a target value on the x-axis or y-axis, so you can generate a target point by specifying its x-y coordinates.



With this program, you can see the value change (displayed on the LED panel) after Smart Camera detects a color block.

In this sample program, the Kp value is set to 0.5. When the target coordinates are specified, the point represented by them serves as the center point.

On the x-axis, the left section ranges from -50 to 0, and the right section from 0 to 50. On the y-axis, positive numbers go upward from the center point, ranging from 0 to 50, and negative ones go downward from the center point, between -50 and 0. The ranges here won't change even when the target coordinates change. It's the corresponding mapping interval that changes.

Take the x axis for an example. If the x coordinate of a point of Smart Camera's frame is set to 100, under the coordinate system of Smart Camera (presented in **BLACK**), the range of the left section is between 0 and 100, and the range of the right section is 100 to 329.

However, under the smaller coordinate system (presented in **RED**), the range of the left section is still between -50 and 0, and the range of the right section is still between 0 and 50. The coordinate system of Smart Camera is unchangeable, therefore, you need Numerical Mapping to keep the values of the small coordinates unchanged. So, the interval [0, 100] in the coordinate system of Smart Camera is mapper to [-50,0] through the formula. The interval of the right section [100, 319] is mapped to [0, 50] through the formula. The Kp value plays a role in changing the intervals of the small coordinate system. For example, if the Kp value is set to 0.8, then intervals for the x and y axis will be between -80 and 80.

 $x_1$  and  $y_1$  are the initial coordinates set through the **Event-specific Smart Camera** block.  $x_2$  and  $y_2$  are the current coordinates of the color block being tracked.  $Z_1$  is the difference between  $x_1$  and  $x_2$ , and  $Z_2$  is the difference between  $y_1$  and  $y_2$ .

$$Z_{1} = \begin{cases} x_{1} > x_{2}, & (x_{2} - x_{1}) \div (x_{1} \div 10 * \text{kp}) \\ x_{2} > x_{1}, & (x_{2} - x_{1}) \div ((319 - x_{1}) \div 10 * \text{kp}) \end{cases}$$

$$Z_{2} = \begin{cases} y_{1} > y_{2}, & (y_{1} - y_{2}) \div (y_{1} \div 10 * \text{kp}) \\ y_{2} > y_{1}, & (y_{1} - y_{2}) \div ((239 - y_{1}) \div 10 * \text{kp}) \end{cases}$$

Use the calculated result to control the motors:

leftSpeed =  $Z_1 + Z_2$ 

rightSpeed =  $-Z_1 + Z_2$ 

You may notice a negative sign is added before Z<sub>1</sub> in the rightSpeed formula. That's because Z<sub>1</sub> is linked to X coordinates, which are used to control in mBot's left-right turnings. When mBot makes turns, the two motors rotate at the same speed in different directions. If the color block is to the left of the target point, then the x coordinate in the smaller coordinate system (in RED) lies in the negative interval. The value is assigned to control the motor for the left wheel and the left wheel will rotate backward. If the same value is assigned to control the right wheel, then mBot will move backward rather than make a turn. That's why the negative sign is added intwo negatives make one positive—to make the right wheel rotate forward. Then, mBot will turn left.



The Kp value affects the mapping intervals, which affect the motion of the motors.



# Lesson 4 Object Recognition III

Subject: Computer Science/Design and Technology Level of Difficulty: Intermediate **Topic:** Visual Recognition **Efforts:** 60 minutes



#### Learning Goals

- How does Smart Camera measure the width and height of a detected color block?
- How to apply the color block detection mode in different scenarios?

# 🗄 Learning Content

#### Distance from point to plane

An image returned by Smart Camera is two-dimensional or flat, so the module can't measure its distance to an object. There's a concept that can help—**angle of view**. You may have such an experience: when you walk close to a thing, it seems bigger because your angle of view changes; when the object is far away, it appears smaller.

This also works on Smart Camera. Smart Camera can measure the width and height of a detected color block and then calculate the area of the color block with the two parameters. The module can calculate its distance to the color block based on the visual perception that closer objects seem bigger and farther ones smaller.







#### **Distance learning**

The detected area of the color block is the key to enabling mBot to automatically keep a distance to the object.

The logic here is: when the detected area is larger than the target area, mBot moves backward; when the detected area is smaller than the target area, mBot moves forward.

You have to be careful not to limit the target area with a specific value. The target area should be within a certain range. If the target area is specified to a value, it causes frequent vibrations because mBot tries to move to reach the value.

Think about this question: If the distance that mBot needs to keep to the object keeps changing, what will you do to enable mBot to automatically learn the distance?

Write down your solutions here:



#### Hands-on

#### 1. Task

Program mBot to move based on the area and coordinates. The program enables mBot to make decisions on whether to move forward or backward based on the width and height and to turn left or right based on the coordinates.

Question 1: How to enable mBot to automatically learn the distance to an object?

**Question 2:** How to enable mBot to track an object based on the coordinates and the area size?

#### 2. Algorithm Design

Use a flowchart to describe the logic of your program. Draw it in the space below:





# ⊞ Quiz

1. When the **Signature 1 range** increases, the inclusiveness of the color block \_\_\_\_\_\_.

A. decreases	B. increases	C. stays the same
--------------	--------------	-------------------

2. In the area-based tracking algorithm, setting a range for the target area helps

A. increase speed B. reduce speed C. improve stability D. reduce stability

# 🛱 Extension

Now mBot can track objects based on area size. Can you add a gripper to the robot? When the object appears in the detection range, mBot tracks it down and grabs it with the gripper.

#### Learn More: Compatibility

PixyMon is not only the tool for you to view images returned by Smart Camera or assistant in teaching the module color blocks but also a useful debugging tool. When Smart Camera has a larger angle of view, it can "see" more. However, seeing more sometimes comes with more disturbances, and as a result, the detected objects aren't the ones you intended. For example, in the figure below, Smart Camera detects the ball but also a packaging bag.



You need to adjust the inclusiveness of the color block to reduce false detections. In PixyMon, Go to **File > Pixy Parameters (saved on Pixy) > Tuning**, then you can adjust the compatibility. **The higher value you choose, the more inclusive of the color block will be.** 

Pixy Parameters (saved on Pixy) PixyMon Parameters (saved on computer)						
Tuning	Expert	Signature La	abels	Camera	Interface	Servo
Signature	e 1 range	3.800000				
Signature	e 2 range	3.500000	-0-			
Signature	e 3 range	3.500000	-0-			
Signature	e 4 range	3.500000	-0-			
Signature	e 5 range	3.500000	-0-			
Signature	e 6 range	3. 500000	-0-			
Signature	e 7 range	3. 500000	-0-			
Min bi	rightness	0.200000	c	0		
Camera bi	rightness	50	-0-			

Note: Light intensity, installation angle, and other factors may also affect the detection process and results. Tune the setting based on the actual environment. A lower value doesn't always bring a more accurate detection result.

# Lesson 5 Line Tracking

Subject: Computer Science/Design and Technology Level of Difficulty: Intermediate **Topic:** Visual Recognition **Efforts:** 60 minutes

#### Learning Goals

- What is the line-tracking feature of Smart Camera?
- How to design an algorithm of visual line tracking?

#### 🗄 🛛 Learning Content

Smart Camera can detect lines and intersections. By clicking **Program > line\_tracking** in PixyMon, you enable Smart Camera's line detection mode. In this mode, all images returned are black-and-white, and PixyMon displays all the lines detected in color.

Smart Camera is able to make decisions when it recognizes intersections. It can tell how many branches there are in the intersection and any intersections that lines form. The red line called **Vector** indicates the line that Smart Camera is currently following.

**Note:** mBot's turn angle keeps changing when moving. Smart Camera may make different decisions even when dealing with the same intersection.



Figure 1 Line Smart Camera's currently following



Figure 2 Following the red line

There are two modes: dark line on light background and light line on dark background.



**Question:** Apart from the map color, are there any other constraints that might influence the module's performance when it's detecting lines?

Let's use Map A included in the package to test your Smart Camera's capability of detecting lines. Use the table below to conduct the experiment to identify the constraints.

	Distance	Angle	Able to identify or not
Solution 1	10cm	Vertical	
Solution 2	10cm	Horizontal	
Solution 3	5cm	Vertical	
Solution 4	5cm	Horizontal	
Solution 5	20cm	Vertical	
Solution 6	20cm	Horizontal	
Conclusion			

**Note:** You can reduce the false detections by configuring the line width in PixyMon.

Go to File > Configure > Pixy Parameters (saved on Pixy) > Tuning to adjust Maximum line width and Minimum line width.

Go to **Action > Toggle lamp** to turn on Smart Camera's LED lamp. Go to the **Color Block Detect** category in mBlock to find the coding block below:



Configure					X
Pixy Param	eters (save	ed on Piny)	PixyMon Param	eters (saved	on computer)
Tuning	Expert	Barcode Lab	els Camera	Interface	Servo
Edge	threshold	35			
Maximum J	line width	100			
Minimum J	line width	0	0		
Y	'hite line				
Camera 1	orightness	50			
			OK	Cancel	Apply



#### Hands-on

#### 1. Task

Program and execute mBot to perform visual line tracking task on Map A.



**Question:** Which of the following parameters set can lead to faster and more accurate line detection? Test your mBot to complete the table below.

	Кр	Target speed	Result
Test 1	0.3	30	
Test 2	0.5	60	
Test 3	0.6	70	
Custom			



#### 2. Algorithm Design

Use a flowchart to describe the logic of your program. Draw it in the space below:





### ⊞ Quiz

- 1. After lowering Smart Camera, the module still fails to detect lines.
- You should adjust ( ).
- A. Maximum line width B. Minimum line width C. Camera brightness
- 2. In the image returned, Smart Camera ( ) lines in face of intersections.
- A. Red B. Yellow C. Green D. Blue

# Extension

The package has a Map B which has thinner lines. Could you use the example program above to have mBot follow lines on Map B as well?





# Lesson 6 Path Selection

Subject: Computer Science/Design and Technology Level of Difficulty: Intermediate **Topic:** Visual Recognition **Efforts:** 60 minutes

#### Learning Goals

- How can mBot do a circular line tracking? How many solutions can you figure out?
- How can mBot identify and choose branches?

### 🗄 Learning Content

A self-driving car needs to detect roadways and determine the best route so as to arrive at the destination. Branch selection is another problem that a self-driving car faces when driving. When it comes to Smart Camera, we know it's able to detect lines and intersections and return the intersection's current coordinate in the image. Therefore, we have reasons to believe that it can help mBot plan its route and get to the destination.





smart camera port1 
detected cross-point?

🖴 smart camera port1 🔻 detected cross point's 🗴 🔻 coordinate

Identifying intersections is one of Smart Camera's essential features. But what action Smart Camera is to take after identifying the intersection? Here are 3 options for reference:

**Option 1:** Use Smart Camera's default feature to identify and select the branch. For instance, when the intersection's y coordinate is larger than a specified value, the motor turns left at power of X%, or turns right at power of X%.

**Option 2:** Use a combination of line tracking and barcode detection. Smart Camera relies on its line detection feature to track lines, and chooses which branch to take based on the barcode placed near the intersection.

**Note:** You can program Smart Camera to make decisions based on barcodes' coordinates, which help avoid repetitive detection.



Where to place barcode labels:

**Option 3:** Have the IR sensor work with Smart Camera. The IR sensor is responsible for detecting lines while Smart Camera decides which branch to take based on its line or barcode detection feature.



#### Hands-on

#### 1. Task

Use the three options to perform line tracking and branch selection on Map B. Compare their pros and cons. Fill in the table below.



	Pros	Cons
Option 1		
Option 2		
Option 3		



#### 2. Algorithm Design

Use a flowchart to describe the logic of your program. Draw it in the space below:





# ⊞ Quiz

- 1. Can Smart Camera identify intersections automatically?
- A. Yes B. No
- 2. Which two modes can work together?
- A. Color block detection and barcode detection
- B. Barcode detection and line tracking
- C. Color block detection and line tracking
- D. None of them

# 🛱 Extension

Put together two pieces of Map B. How could mBot track lines on this scenario?



# Mini Capstone <u>Automatic Sorting</u> Machine

Subject: Computer Science/Design and Technology Level of Difficulty: Intermediate **Topic:** Visual Recognition **Efforts:** 120 minutes

# \*

#### Learning Goals

- How can automation and Intelligent machines increase the efficiency of waste collecting and sorting?
- How to design and construct a serve robotic arm?

#### Earning Content

Effective waste management contributes to resources recycling and reduces environmental pollution. One of the challenges is how to sort different types of waste that affect the efficiency of recycling. Engineers in Visual Lab plan to design an automatic sorting machine to make the sorting process easier. This machine is expected to automatically collect and sort garbage.

To prepare for the future, engineers in Visual Lab are considering automatic waste sorting system. In a world where waste sorting becomes an integral part of human's life, autonomous garbage trucks are expected to automatically drive to delegated locations to collect garbage, saving us much trouble and time.

#### Performance Task 1

Question 1: How can the robot "pick" the garbage?

A self-driving garbage truck could be a good application. The truck automatically drives to a specified destination to fetch the garbage for recycling.

In the demonstration below, it uses balls or dots to represent the garbage. put the balls anywhere in Area I (see, e.g., the green and orange balls). mBot is in Area II. How can mBot fetch the balls from Area I to Area II?



**Question 2:** How do you enable the truck to automatically drive to a designated location to fetch the garbage (the ball)?





Servo, also called servo motor, is a type of motors. It's widely used in robotic arms, humanoid robots and more due to its precise control of angular motion.





Components of the servo: servo + servo hub + acrylic structural member.



The servo must work with an RJ25 adapter. You can connect the servo to the corresponding slots on the adapter. The servo included in the package can turn to max 180°.



Find the coding block to program a servo: go to **Extension > Maker's Platform**, and find the block at the **Maker's Platform** category.

눩 servo port1 ▼ slot1 ▼ positioned at 90 angle

#### Example:



#### Performance Task 2

You've designed a robotic arm. Now plan the route, and use the self-driving technology to automatically fetch garbage at designated locations.

The garbage truck sets out from the Start Point. It drives past A, B, and C to collect the garbage (the balls) and takes them back to a designated location that is assigned by you.





#### Question: What are the routes for collecting garbage?

Plan a route for mBot to collect the garbage at A, B, and C. Put barcode labels at intersections to help mBot successfully fetch the balls.

Place each barcode label at where it belongs based on your route plan.

Complete the table below to write down your solution.

Design Specification				
Fetch				
Line following				
Robotic arm				
Issues				





# ⊞ Quiz

1. What's the 9g servo's max turn angle?

A. 90° B. 180° C.270° D.360°