



This is the beta version tutorial. It may have some issues. But please do not worry about it. We will continuously update it. Besides, your idea or feedback are all welcome. You can send your idea or feedback via the comments on the [update #12](#). Thanks.

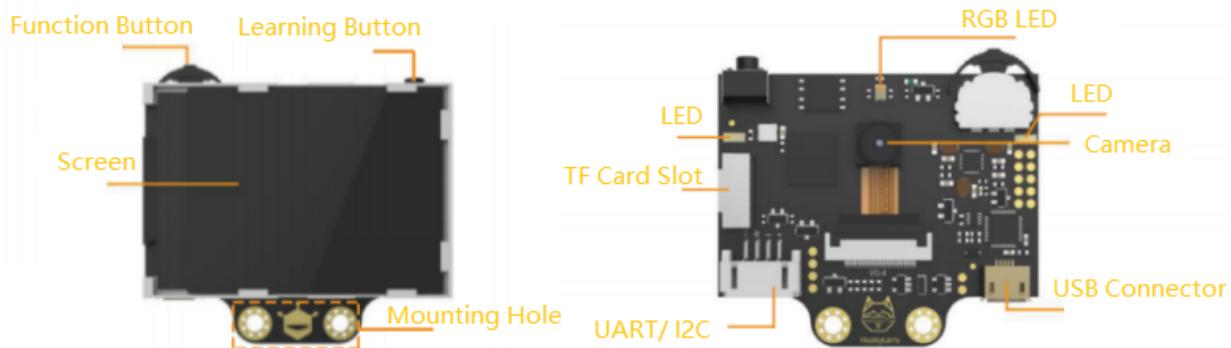
Introduction

HuskyLens is an easy-to-use AI vision sensor. It has 6 built-in functions: Face Recognition, Object Tracking, Object Recognition, Line Tracking, Color Recognition and Tag Recognition. Through the UART / I2C port, HuskyLens can connect to Arduino, Raspberry Pi, or micro:bit, and make your very creative projects without playing with complex algorithms.

Specification

- Processor: Kendryte K210
- Image Sensor:
 - SEN0305 HuskyLens: OV2640 (2.0Megapixel Camera)
 - SEN0336 HuskyLens PRO: OV5640 (5.0MegaPixel Camera)
- Supply Voltage: 3.3~5.0V
- Current Consumption: 320mA@3.3V, 230mA@5.0V (face recognition mode; 80% backlight brightness; fill light off)
- Connection Interface: UART; I2C
- Display: 2.0-inch IPS screen with 320*240 resolution
- Built-in Algorithms: Face Recognition, Object Tracking, Object Recognition, Line Tracking, Color Recognition, Tag Recognition
- Dimension: 52mm * 44.5mm

Board Overview



- Function Button: Dial left or right to switch functions
- Learning Button: Point at the object to be identified, press the learning button to learn the specified object

- USB Connector: connect to the power supply; connect to the computer to upgrade the firmware
- 4pin Connector in UART Mode

Num	Label	Pin Function	Description
1	T	TX	TX pin of HuskyLens
2	R	RX	RX pin of HuskyLens
3	-	GND	negative pole of power supply(0V)
4	+	VCC	positive pole of power supply(3.3~5.0V)

- 4pin Connector in I2C Mode

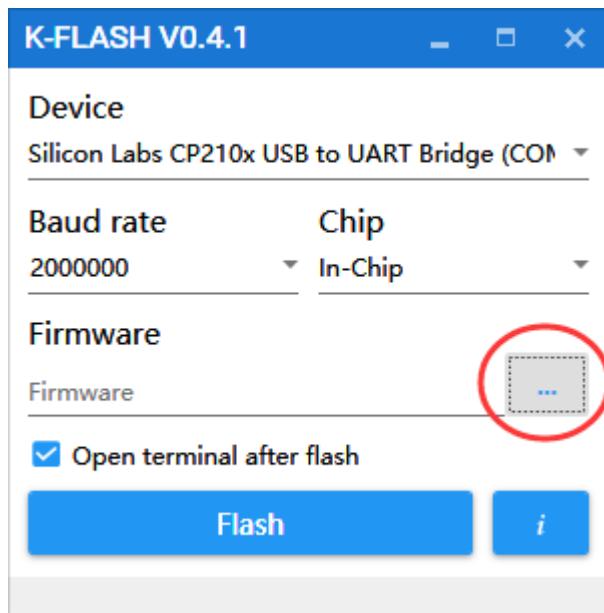
Num	Label	Pin Function	Description
1	T	SDA	serial clock line
2	R	SCL	serial data line
3	-	GND	negative pole of power supply(0V)
4	+	VCC	positive pole of power supply(3.3~5.0V)

Upgrade Firmware

Before using this product, it is strongly recommended that you upgrade HuskyLens to the latest version of the firmware, especially for Kickstarter backers.

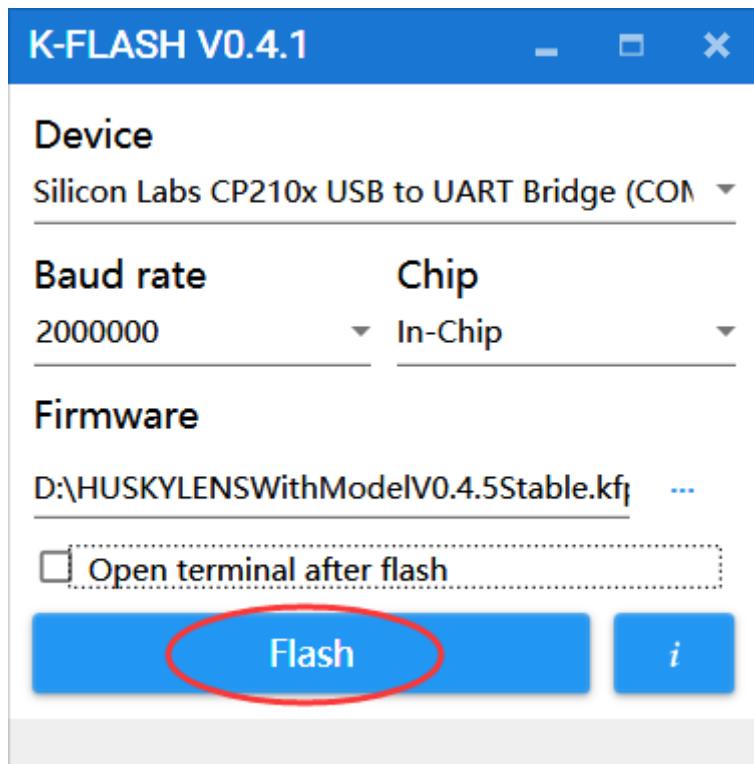
The steps are shown as follows:

1. Download the K-Flash software. [Click here](#) to download it. Currently it only supports windows operating system.
2. Download the USB to UART driver, and install it.[Click here](#) to download it. HuskyLesn adopts the CP2102 chip to implement the USB to serial port function.
3. Download the latest firmware. [Click here](#) to view all versions of firmware. In this tutorial, we adopts this firmware: **HUSKYLENSWithModelV0.4.5Stable.kfpkg**.
4. Open the K-Flash software, then click the button(...) to load the new firmware.

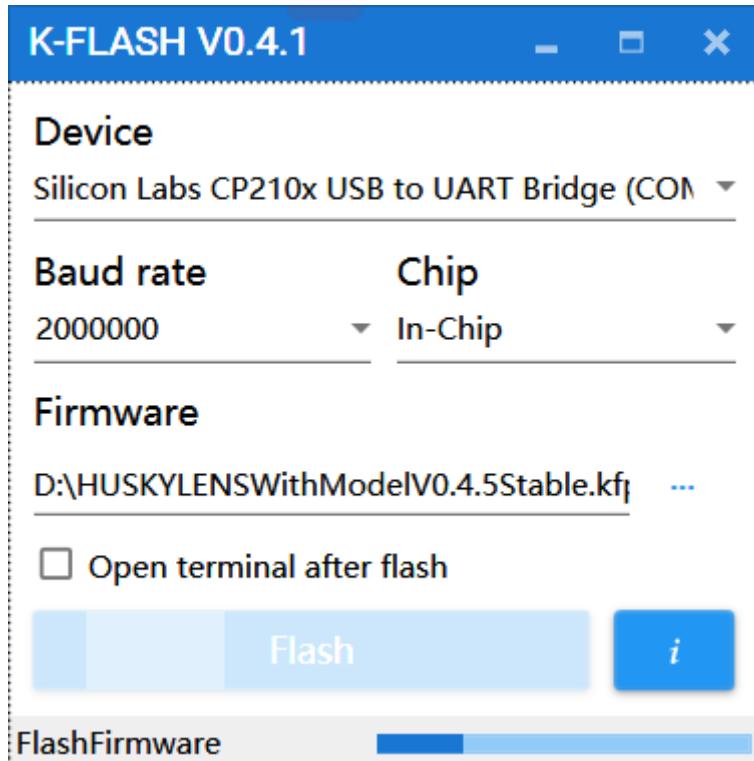


5. Follow the parameters below to set the K-Flash software:

- Device: the COM port on your computer
- Baud rage: 2000000
- Chip: In-Chip
- Not tick 'Open terminal after flash'



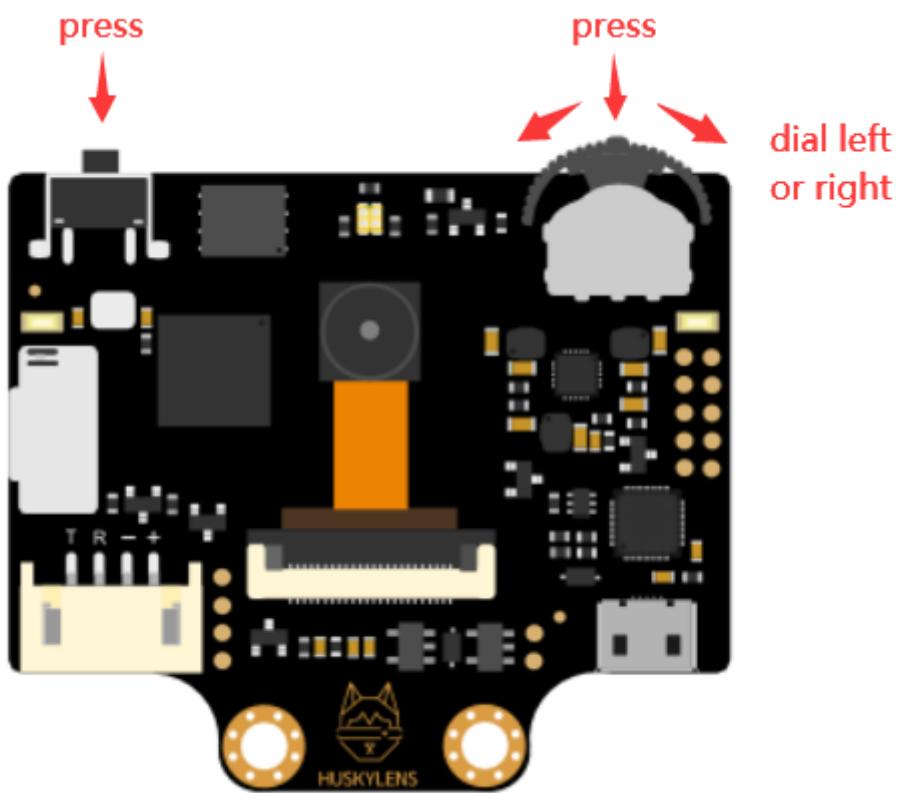
6. Click the Flash button. Wait about 5 minutes to complete the uploading. This is a major upgrade and therefore it takes a long time.



7. Upgrade has been completed now. Enjoy it.

Function Button and Learning Button

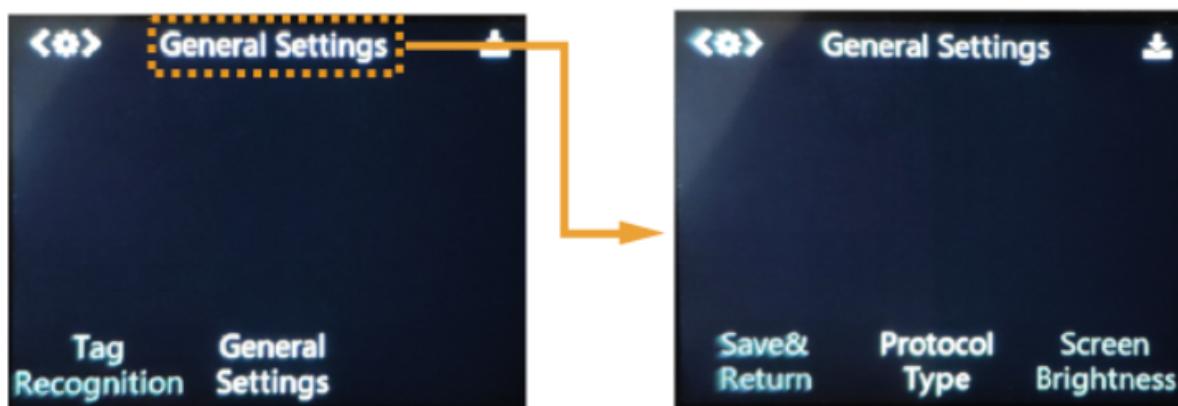
- Dial the function button left or right to select different functions;
- short press the learning button to learn the specified object; long press the learning button for continuous learning, that is, learning the object from different angles and ranges when you adjust the distance and angle from the object and HuskyLens
- Long press the function button to enter the parameter setting of the current function; Then, dial the function button left or right to select different parameter, short press the function button to set each parameter, dial the function button left or right to adjust the parameter.



General Settings

There are 10 different parameter settings in the general settings. The operation shows as follows:

1. Select the general settings: Dial the function button right to select it, then the words "General Settings" is displayed at the top of the screen.
2. Enter the general setting mode: Short press the function button to enter it.



3. Dial the function button left or right to select different parameter, then short press the function button to set the parameter. Dial the function button left or right to adjust the parameter. Then short press the function button again to confirm the parameter.

4. Save the settings: After adjusting the parameters, dial the function button left, to select "Save & Return", then short press the function button. A message "Do you save data?" will appear. The default selection is "Yes". At this time, short press the function button to save and exit.



Introducing each parameter:

- **Protocol Type**

Huskylens supports three UART baud rates (9600, 115200, 1000000), and also supports I2C protocol. In addition, it supports auto detection protocols, that is, it will automatically switch between UART and I2C. We recommend to use the auto detection protocol, which is simple and convenient. The default value is auto detect.

- **Screen Brightness**

The screen supports different brightness settings. The brightness setting value is 1 ~ 100. The default value is 80.

- **Menu Auto Hide**

When you don't operate the Huskylens for a period of time, the menu on the screen will automatically hide. You can adjust this period. The value is 1-100 seconds. The default value is 10 seconds.

- **LED Switch**

There are two LED lights on the front of the Huskylens. You can set the ON and OFF status of these LED lights. The default value is OFF.

- **LED Brightness**

You can set the brightness of these two LED light, the brightness range is 1 ~ 100. The default value is 50.

- **RGB Switch**

There is also an RGB light on the front of the Huskylens. You can set the ON and OFF status of the RGB light. The default value is ON.

- **RGB Brightness**

The brightness of this RGB light is also adjustable, with a brightness range of 1--100. The default is 20.

- **Factory Reset**

Huskylens will be reseted to factory settings via this function.

- **Version**

The current version of the built-in firmware.

- **Language**

Huskylens supports Chinese and English language. You can choose your favourite language.

Face Recognition

1. Learn a single face

The default setting is to learn a single face.

Operation and Setting

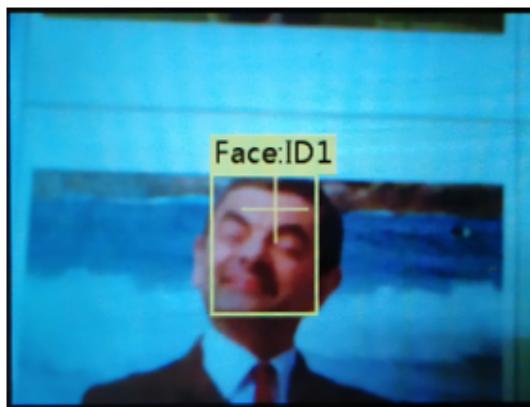
Dial the function button to the left until the word "Face recognition" is displayed at the top of the screen.

Learning and Detection

1. **Face Detection:** Point the HuskyLens at any face. When a face is detected, it will be automatically selected by a white frame with words "Face" on the screen.

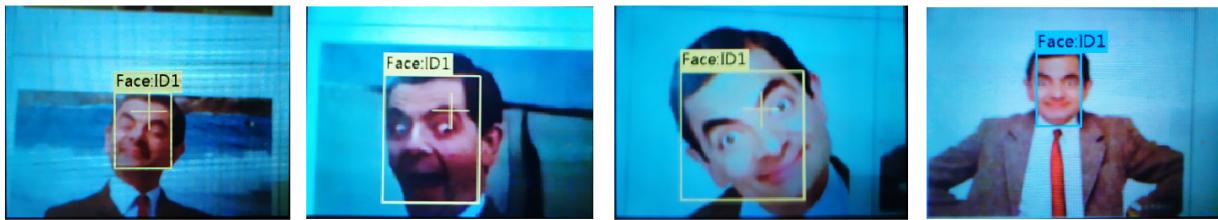


2. Face Learning: Point the "+" symbol at the face, short press the "learning button" to learn the face. Then a blue frame with words "Face: ID1" will be displayed if HuskyLens detects the same face, which indicating that HuskyLens has learned the face and can recognize it now.



However, HuskyLens only learned one plane (one-dimensional) of the face. But a face is three-dimensional. If the angle of the face has been changed, HuskyLens may not recognize it. So you need to let HuskyLens learn from different angles of the face of the same person. The operation shows as follows: (Before HuskyLens learning news thing, please let HuskyLens forget old thing first.)

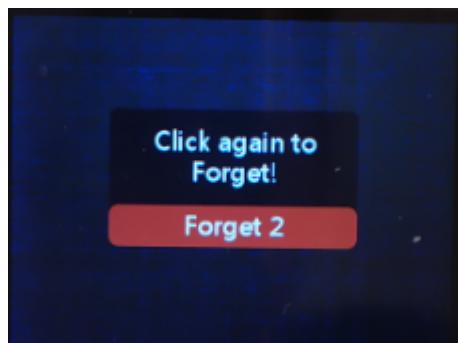
Keep pressing the "learning button", point HuskyLens at different angles of the face. During this process, a yellow frame with words "Face: ID1" will be displayed on the face, which indicates HuskyLens is learning the face. After HuskyLens learned all angles of the face, release the "learning button". A blue frame with words "Face: ID1" will be displayed, indicating that learning of face information is completed. Now HuskyLens can recognize different angles of the face of the same person.



Tips: If there is no "+" symbol in the center of the screen before learning, it means that the HuskyLens has already learned, now HuskyLens is detecting face. If you want to let HuskLens learn new face, you need to let HuskyLens forget the learned face first.

3. **Face Recognition** The learned face information is automatically saved. When HuskyLens detects the learned face from multiple faces, the learned face will be selected with a blue frame and identify face: ID1.
4. **Forget the Learned Face** If you want to recognize another face, or re-enter face information, you need to delete the current face information. When HuskyLens is in the face recognition mode, short press the "learn button", the screen displays "click again to forget". Before the countdown ends, short press the "learning button" again to delete the learned face information, then the yellow "+" symbol is displayed. Now you can let HuskyLens learn a new face.

In other functions, the operation of forgetting is the same. It will not be repeated in subsequent chapters.



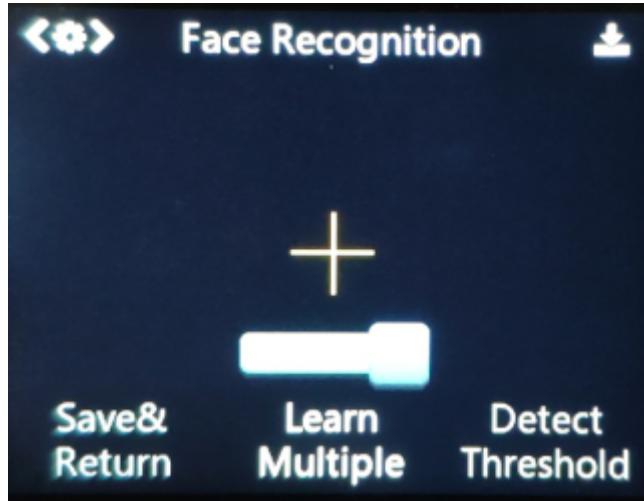
2. Learn multiple faces

The default setting is to learn a single face. In order to learn multiple faces, we need to enable "Learn Multiple" of face recognition.

Operation and Setting

1. Dial the function button to the left until the word "Face recognition" is displayed at the top of the screen.
2. Long press the function button to enter the parameter setting of the face recognition function.

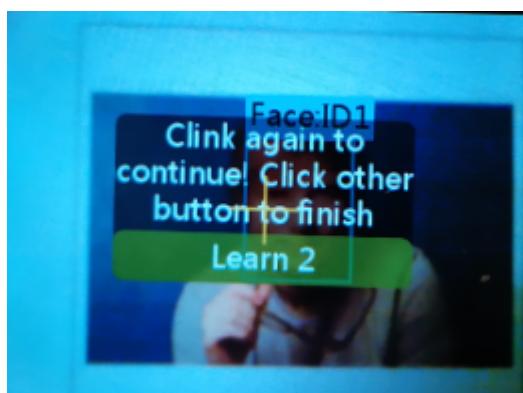
3. Dial the function button until select "Learn Multiple", then short press the function button, and dial to the right to turn on the "Learn Multiple" switch, that is, the square icon on the progress bar is turned to the right. Then short press the function button to confirm this parameter.



4. Dial the function button left until select "Save & Return", and short press the function button to save the parameters and return automatically.

Learning and Detection

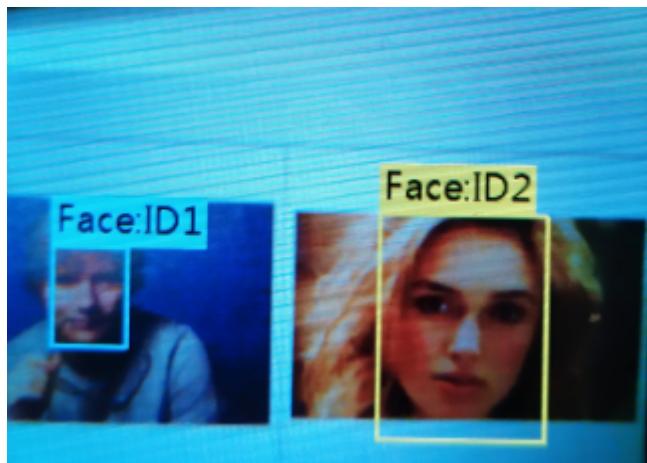
1. **Multiple Faces Learning:** Point the "+" symbol at the face, short or long press the "learning button" to learn the face of the first person. Then release the "learning button", a blue frame with words "Face: ID1" will be displayed if HuskyLens detects the same face, meanwhile, a message "Click again to continue! Click other button to finish" will be displayed. Please short press the "learning button" before the countdown ends if you want to learn the face of other person. If not, short press the "function button" before the countdown ends, or do not press any button to let the countdown ends.



In this chapter, we want to continuously learn other person's faces. So we need to short press the "learning button" before the countdown ends. Then we can let HuskyLens learn the face of the second person. It will be identified as "Face: ID2" .

Tips: If there is no "+" symbol in the center of the screen before learning, it means that the HuskyLens has already learned, now HuskyLens is detecting face. If you want to let HuskyLens learn new face, you need to let HuskyLens forget the learned face first.

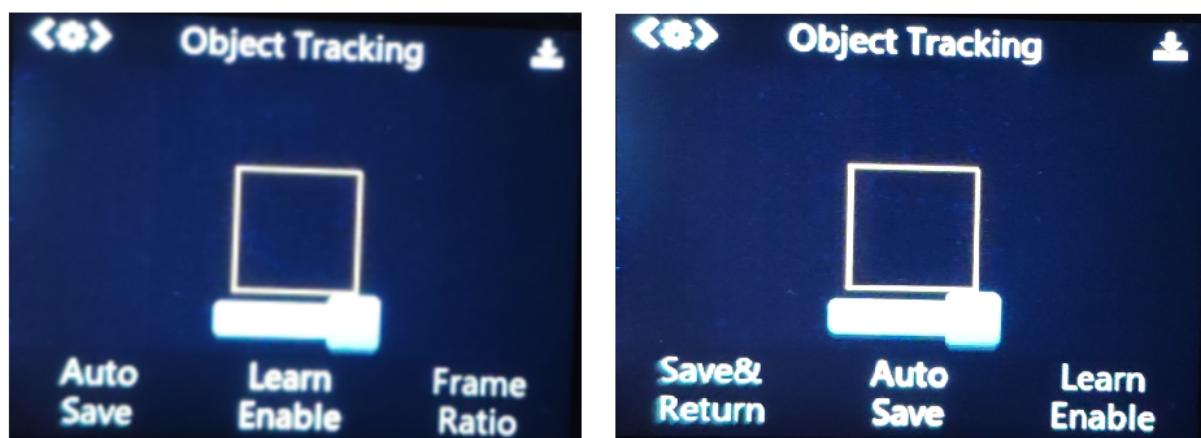
- 3. Multiple Faces Recognition** The learned face information is automatically saved. When HuskyLens detects the learned face of the first person from multiple faces, the learned face will be selected with a blue frame and identify face: ID1; When HuskyLens detects the learned face of the second person, it will be selected with a blue frame and identify face: ID2; and so on.



Object Tracking

Operation and Setting

1. Dial the function button to the left or right until the word "Object Tracking" is displayed at the top of the screen.
2. Long press the function button to enter the parameter setting of the object tracking function.
3. Dial the function button right until select "Learn Enable", then short press the function button, and dial to the right to turn on the "Learn Enable" switch, that is, the square icon on the progress bar is turned to the right. Then short press the function button to complete this parameter.
4. As above, please enable "Auto Save" parameter.



5. You can also adjust the size of the frame by setting "Frame Ratio" and "Frame Size" to match the shape of the object.
6. Dial the function button left until select "Save & Return", and short press the function button to save the parameters and return automatically.

Learning and Detection

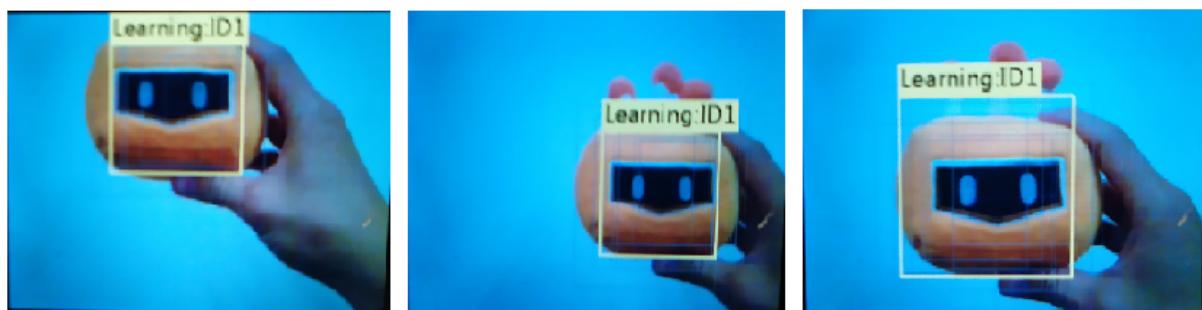
1. Object Learning

By adjusting the distance from the Huskylens and the object, the object to be tracked should be included in the yellow frame in the center of the screen. Then long press "learning button" to learn the object from various angles and distances. During the learning process, the yellow frame with words "Learning: ID1" will be displayed on the screen. Then release the "learning button" to complete the learning.



2. Object Tracking

Move the HuskyLens or the target, the frame will automatically track the target. When tracking the object, the yellow words "Learning: ID1" will be displayed, indicating that HuskyLens is tracking the object while learning. This character helps improve the object tracking ability.



Tips:

- Only one object can be tracked at a time. It can be any object with a clear outline, even various gestures.
- If there is no yellow frame on the center of the screen, it means that the HuskyLens has already learned a object. Please refer to the method of delete faces in face recognition, to delete the learned object.

Object Recognition

HuskyLens can recognize 20 built-in objects. They are aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, diningtable, dog, horse, motorbike, person, pottedplant, sheep, sofa, train, tvmonitor.

1. Recognize a single object

The default setting is to recognize a single object.

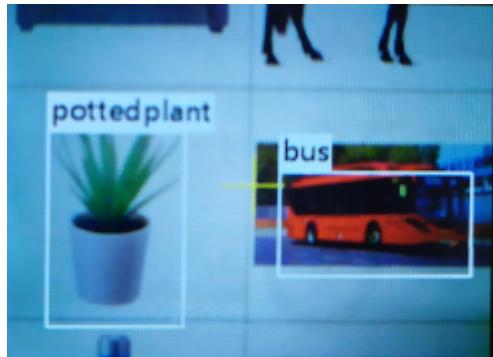
Operation and Setting

Dial the function button to the left until the word "Object Recognition" is displayed at the top of the screen.

Learning and Detection

1. Object Detection

When detecting objects, HuskyLens will automatically recognize it, and the object will be displayed by the white frame with its name on the screen. At present, only 20 built-in objects can be recognized, and the remaining objects cannot be recognized temporarily.



2. Object Mark

Point the "+" symbol at the object, then short press the "learning button". When pressing, the color of the frame changes from white to blue, and the name of the object and its ID number will appear on the screen.

Tips: If there is no yellow "+" symbol on the center of the screen, it means that the HuskyLens has already learned an object. Please refer to the method of delete faces in face recognition, to delete the learned object.

3. Object Recognition

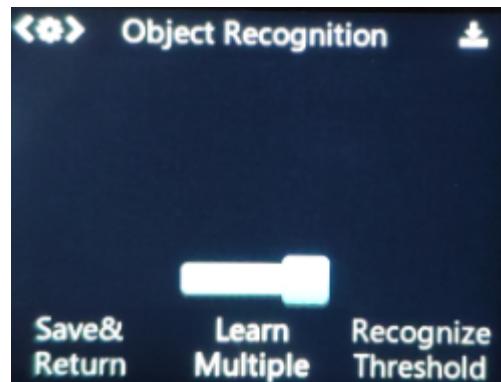
When encountering the same objects, they will be selected by the blue frame and the name and ID number will be displayed. When encountering other ones, they still with a white

frame selection. This can be used as a simple filter to find out what you need from a bunch of objects.

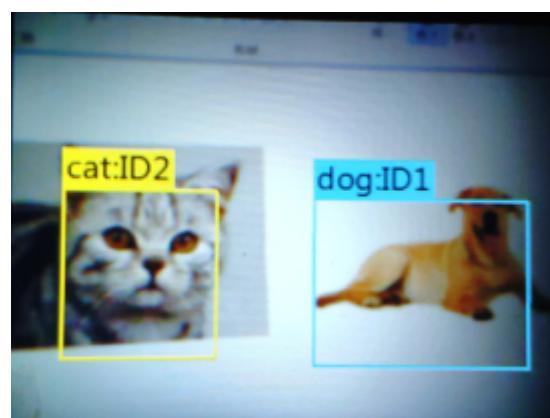


2. Recognize multiple objects

The default setting is to recognize a single object, so we need to enable "Learn Multiple" parameter of the object recognition function. Please refer to the multiple faces learning and recognition chapter for setting, this chapter will not repeat it.



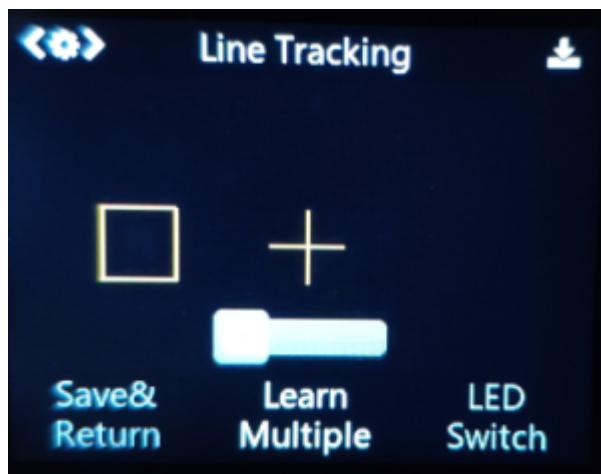
The ID number is related to the order of marking objects. For example, if a dog is marked for the first time and a cat is marked for the second time, when the dog is recognized, the words "dog: ID1" will be displayed on the screen, and when the cat is recognized, the words "cat: ID2" will be displayed on the screen.



Line Tracking

Operation and Setting

1. Dial the function button to the left or right until the word "Line Tracking" is displayed at the top of the screen.
2. Long press the function button to enter the parameter setting of the line tracking function.
3. Dial the function button right until select "Learn Multiple", then short press the function button, and dial to the left to turn off the "Learn Multiple" switch, that is, the square icon on the progress bar is turned to the left. Then short press the function button to complete this parameter.



4. You can also turn on the LEDs by setting "LED Switch". This is very useful in the dark environment.
5. Dial the function button left until select "Save & Return", and short press the function button to save the parameters and return automatically.

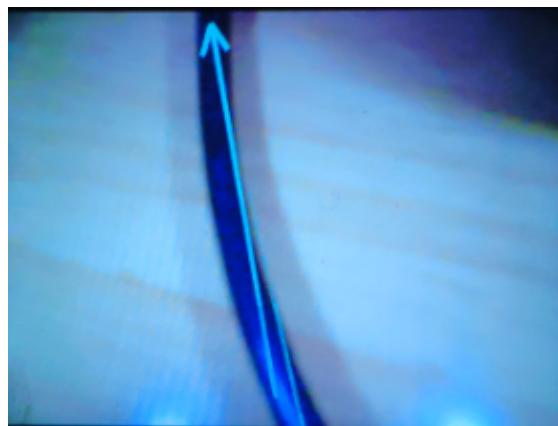
Learning and Detection

1. **Line Learning** Point the "+" symbol at the line, then point the yellow frame at the background area. It is recommended that within the view field, there is only one line that need to be learned and no crossing lines. Then, short press the "learning button" to complete the learning process. A blue route direction arrow will appear on the screen.



Tips: If there is no yellow frame and "+" symbol on the center of the screen, it means that the HuskyLens has already learned a line. Please refer to the method of delete faces in face recognition, to delete the learned line.

2. **Line Prediction** When HuskyLens detects the line which has been learned, a blue arrow will appear automatically on the screen. The direction of the arrow indicates the direction of the line prediction.



Tips:

- When learning the line, we need to adjust the position of HuskyLens to be parallel to the line.
- Huskylens can learn multiple lines according to the color of lines, but these lines must be monochrome lines with obvious color difference from the background.
- In most cases, only one line is tracked. In order to ensure stability, we recommend to track the single line.

Color Recognition

1. Learn a single color

The default setting is to learn a single color.

Operation and Setting

Dial the function button to the right until the word "Color Recognition" is displayed at the top of the screen.

Learning and Detection

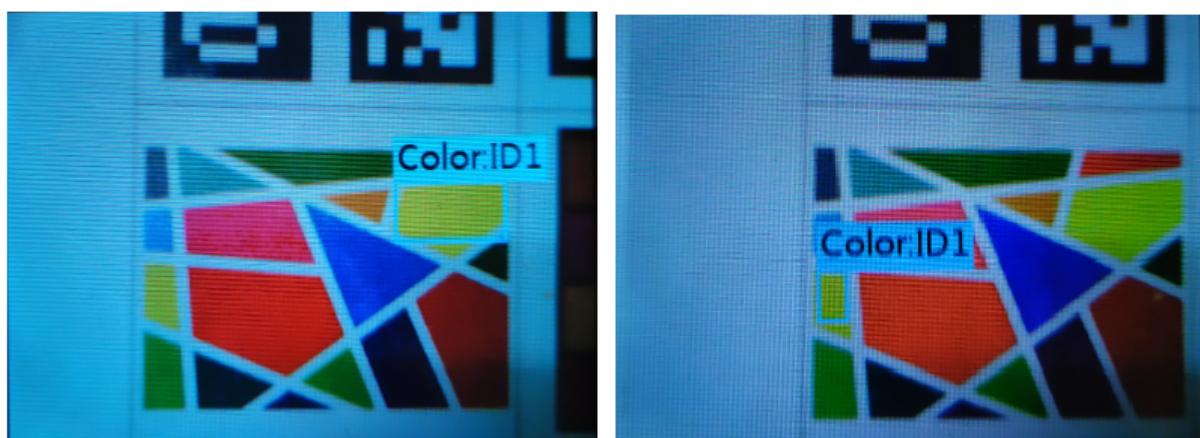
1. **Color Learning:** Point the "+" symbol at the color block, and long press the "learning button". A yellow frame is displayed on the screen, indicating that HuskyLens is learning the color. At this time, adjust the distance and angle between HuskyLens and the color block, to make the size of yellow frame same as the color block. Then, release the "learning button" to complete the learning.



Tips: If there is no "+" symbol in the center of the screen before learning, it means that the HuskyLens has already learned, now HuskyLens is detecting color. If you want to let HuskLens learn new color, you need to let HuskyLens forget the learned color first.

2. **Color Recognition** When encountering the same or similar color blocks, a blue frame with an ID will be automatically displayed on the screen, and the size of the blue frame is same as the size of the color blocks.

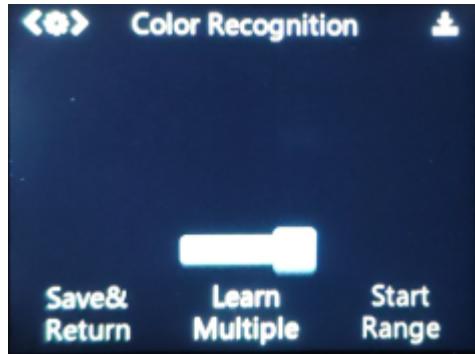
When there are multiple same or similar color blocks appear at the same time , the other color blocks cannot be recognized, that is, only one color block can be recognized at a time.



Tips: Color recognition is greatly affected by ambient light. Sometimes HuskyLens may misidentify similar colors. It is recommended to keep the ambient light unchanged.

2. Learn multiple colors

The default setting is to recognize a single color, so we need to enable "Learn Multiple" parameter of the color recognition function. Please refer to the multiple faces learning and recognition chapter for setting, this chapter will not repeat it.



The ID number is related to the order of learned color. For example, if a yellow block is marked for the first time and a green block is marked for the second time, when the yellow block is recognized, the words "Color: ID1" will be displayed on the screen, and when the green block is recognized, the words "Color: ID2" will be displayed on the screen.



Tag Recognition

1. Learn a single tag

The default setting is to learn a single tag.

Operation and Setting

Dial the function button to the right until the words "Tag Recognition" is displayed at the top of the screen.

Learning and Detection

Please [click here](#) to download the tag / QR code pictures.

1. Tag Detection:

When Huskylens detects the tag, the tag will be automatically selected by the white frame on the screen.



2. **Tag Learning:** Point the "+" symbol at the tag, and press the "learning button". A yellow frame with words "Tag:ID1" will be displayed on the screen, which indicating that HuskyLens is learning the tag now. Then, release the "learning button" to complete the learning process.



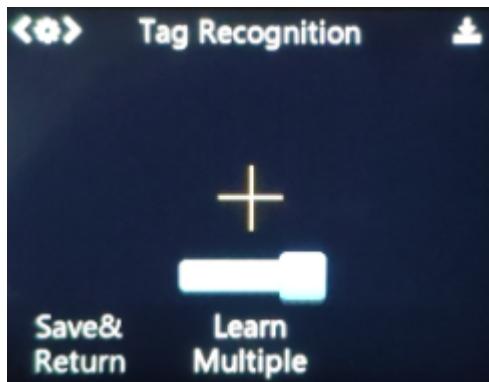
Tips: If there is no "+" symbol in the center of the screen before learning, it means that the HuskyLens has already learned, now HuskyLens is detecting tag. If you want to let HuskLens learn new tag, you need to let HuskyLens forget the learned tag first.

3. **Tag Recognition** When encountering the learned tag, a blue frame with an ID will be automatically displayed on the screen.

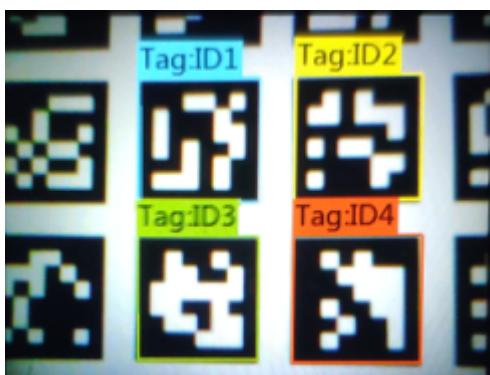


2. Learn multiple tags

The default setting is to recognize a single tag, so we need to enable "Learn Multiple" parameter of the tag recognition function. Please refer to the multiple faces learning and recognition chapter for setting, this chapter will not repeat it.



Similarly, the ID is related to the order of learned tags.



Project Demo(Arduino)

In this chapter, HuskyLens will be connected to Arduino mainboard. Arduino will read data from HuskyLens.

We recommend your HuskyLens adopt the auto detect protocol, which is easy-to-use and convenient.

Requirements

- **Hardware**
 - [DFRduino UNO R3](#) (or similar) x 1
 - [HUSKYLENS](#) x 1
 - M-M/F-M/F-F Jumper wires
- **Software**
 - [Arduino IDE](#)(version 1.8.x is recommended)
 - Download and install the [HUSKYLENS Library](#). ([About how to install the library?](#))

Tips: How to install the library?

Unzip the file, then copy the folder to the "libraries" folder of the Arduino IDE. Then check whether the folder name is "HUSKYLENS". If not, please change it as "HUSKYLENS".

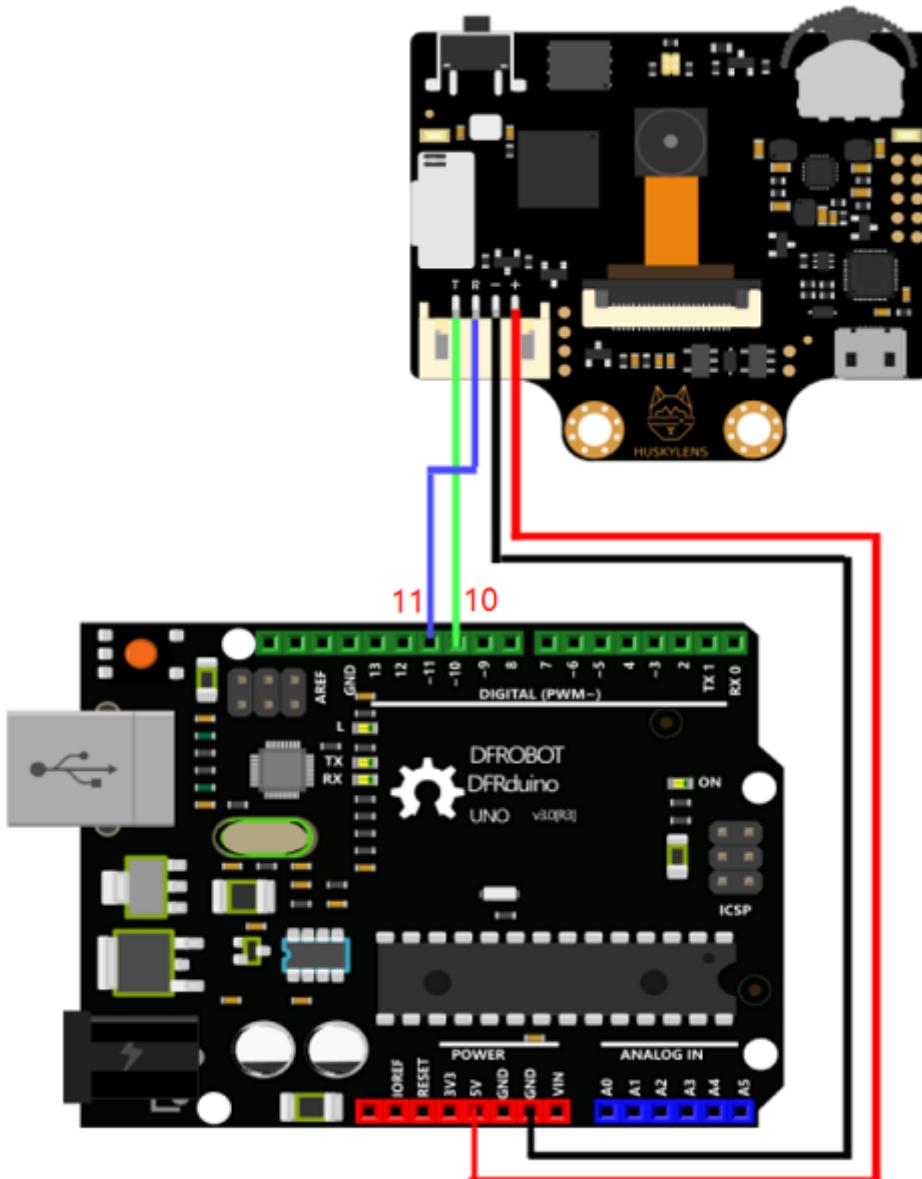
This PC > Local Disk (D:) > arduino-1.8.2 > libraries				
Name	Date modified	Type	Size	
GravityRtc	10/6/2019 4:01 PM	File folder		
GravityTDS	1/3/2018 4:59 PM	File folder		
GSM	10/6/2019 4:01 PM	File folder		
HUSKYLENS	1/23/2020 12:11 PM	File folder		
Keyboard	10/6/2019 4:01 PM	File folder		
LiquidCrystal	10/6/2019 4:01 PM	File folder		
Metro	5/5/2018 4:06 AM	File folder		
Mouse	10/6/2019 4:01 PM	File folder		
OneWire	10/6/2019 4:01 PM	File folder		

All .h files and .cpp files are all in the root directory of the "HUSKYLENS" folder.

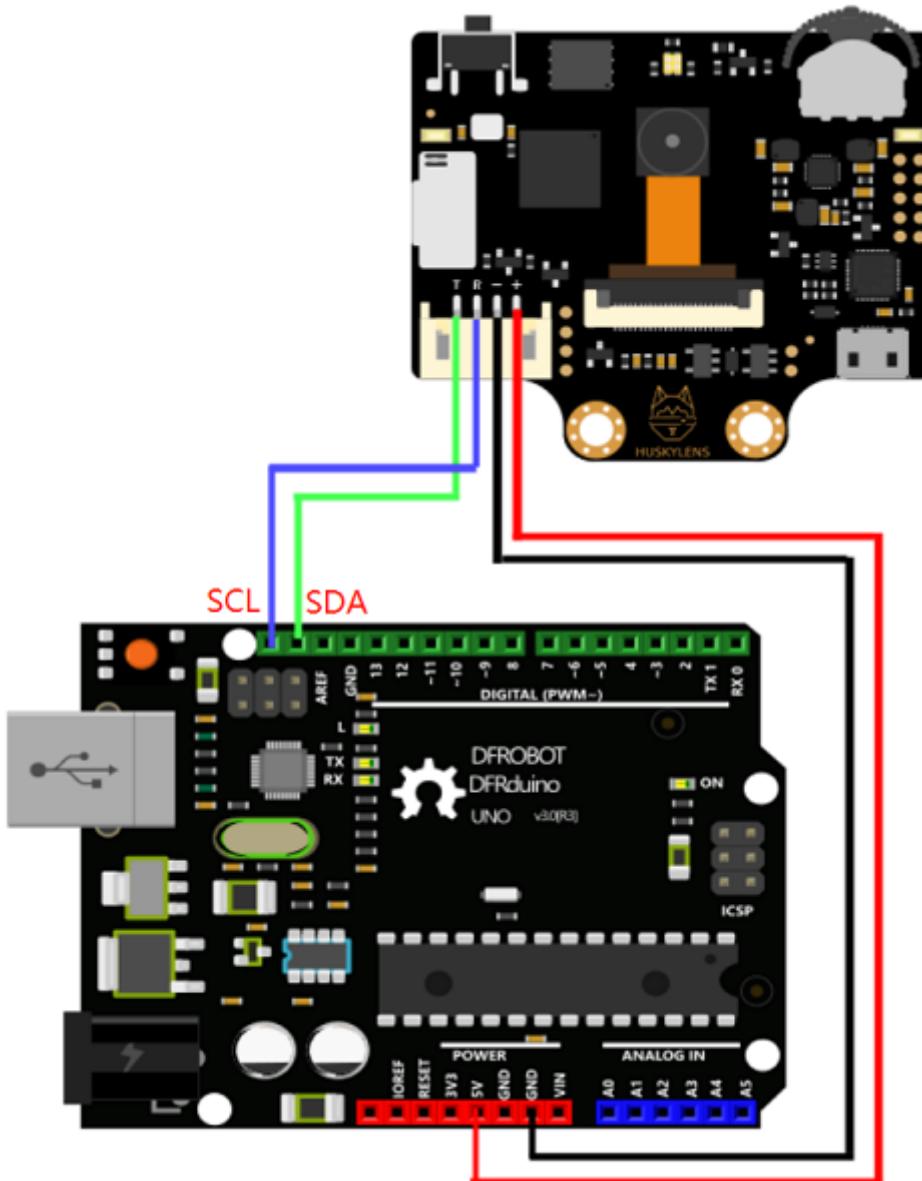
This PC > Local Disk (D:) > arduino-1.8.2 > libraries > HUSKYLENS				
Name	Date modified	Type	Size	
.vscode	1/23/2020 12:11 PM	File folder		
examples	1/23/2020 12:11 PM	File folder		
DFMobile.cpp	1/23/2020 12:11 PM	CPP File	2 KB	
DFMobile.h	1/23/2020 12:11 PM	H File	1 KB	
HUKEYLENS.cpp	1/23/2020 12:11 PM	CPP File	1 KB	
HUKEYLENS.h	1/23/2020 12:11 PM	H File	19 KB	
HUKEYLENSMindPlus.cpp	1/23/2020 12:11 PM	CPP File	1 KB	
HUKEYLENSMindPlus.h	1/23/2020 12:11 PM	H File	7 KB	
HuskyLensProtocolCore.c	1/23/2020 12:11 PM	C File	6 KB	
HuskyLensProtocolCore.h	1/23/2020 12:11 PM	H File	1 KB	
PIDLoop.h	1/23/2020 12:11 PM	H File	3 KB	

Connection Diagram

- **UART Mode(SoftwareSerial)**



- **I2C Mode**



Sample Code

- **UART Mode**

```
#include "HUKSYLENS.h"
#include "SoftwareSerial.h"

HUKSYLENS huskylens;
SoftwareSerial mySerial(10, 11); // RX, TX
//HUKSYLENS green line >> Pin 10; blue line >> Pin 11

void setup() {
    Serial.begin(115200);
    mySerial.begin(9600);
    while (!huskylens.begin(mySerial))
    {
        Serial.println(F("Begin failed!"));
        Serial.println(F("1.Please recheck the \"Protocol Type\" in HUKSYLENS (General Settings>>Protocol Type>>Serial 9600)"));
        Serial.println(F("2.Please recheck the connection."));
    }
}
```

```

        delay(100);
    }
}

void loop() {
    if (!husylens.request()) Serial.println(F("Fail to request objects from
HUSKYLENS!"));
    else if(!husylens.isLearned()) Serial.println(F("Object not learned!"));
    else if(!husylens.available()) Serial.println(F("Object disappeared!"));
    else
    {
        HUSKYLENSResult result = husylens.read();
        if (result.command == COMMAND_RETURN_BLOCK)
        {

Serial.println(String()+F("Block:xCenter=")+result.xCenter+F(",yCenter=")+result.yCenter+
F(",width=")+result.width+F(",height=")+result.height);
        }
        else if (result.command == COMMAND_RETURN_ARROW)
        {

Serial.println(String()+F("Arrow:xOrigin=")+result.xOrigin+F(",yOrigin=")+result.yOrigin+
F(",xTarget=")+result.xTarget+F(",yTarget=")+result.yTarget);
        }
        else
        {
            Serial.println("Object unknown!");
        }
    }
}

```

- **I2C Mode**

```

#include "HUKEYLENS.h"
#include "SoftwareSerial.h"

HUKYLENS husylens;
//HUKYLENS green line >> SDA; blue line >> SCL

void setup() {
    Serial.begin(115200);
    Wire.begin();
    while (!husylens.begin(Wire))
    {
        Serial.println(F("Begin failed!"));
        Serial.println(F("1.Please recheck the \"Protocol Type\" in HUKYLENS (General
Settings>>Protocol Type>>I2C)"));
        Serial.println(F("2.Please recheck the connection."));
        delay(100);
    }
}

void loop() {
    if (!husylens.request()) Serial.println(F("Fail to request objects from
HUSKYLENS!"));
    else if(!husylens.isLearned()) Serial.println(F("Object not learned!"));

```

```

else if(!huskylens.available()) Serial.println(F("Object disappeared!"));
else
{
    HUSKYLENSResult result = huskylens.read();
    if (result.command == COMMAND_RETURN_BLOCK)
    {

Serial.println(String() + F("Block:xCenter=") + result.xCenter + F(",yCenter=") + result.yCenter +
F(",width=") + result.width + F(",height=") + result.height);
    }
    else if (result.command == COMMAND_RETURN_ARROW)
    {

Serial.println(String() + F("Arrow:xOrigin=") + result.xOrigin + F(",yOrigin=") + result.yOrigin +
F(",xTarget=") + result.xTarget + F(",yTarget=") + result.yTarget);
    }
    else
    {
        Serial.println("Object unknown!");
    }
}
}

```

Expected Results

1. Let your HuskyLens learn a new thing first. You can refer to the previous chapters of this tutorial.
2. Then you will get the data.

If HuskyLens is in the line tracking mode, you will get the results like follows:

```

COM4
发送
Arrow:xOrigin=304,yOrigin=158,xTarget=216,yTarget=82
Arrow:xOrigin=312,yOrigin=164,xTarget=224,yTarget=82
Arrow:xOrigin=312,yOrigin=164,xTarget=216,yTarget=86
Arrow:xOrigin=304,yOrigin=164,xTarget=216,yTarget=86
Arrow:xOrigin=304,yOrigin=166,xTarget=216,yTarget=86
Arrow:xOrigin=304,yOrigin=164,xTarget=208,yTarget=86
Arrow:xOrigin=304,yOrigin=164,xTarget=216,yTarget=84
Arrow:xOrigin=312,yOrigin=164,xTarget=216,yTarget=84
Arrow:xOrigin=312,yOrigin=164,xTarget=216,yTarget=84
Arrow:xOrigin=304,yOrigin=164,xTarget=216,yTarget=82
Arrow:xOrigin=312,yOrigin=164,xTarget=216,yTarget=84
Arrow:xOrigin=312,yOrigin=162,xTarget=216,yTarget=82
Arrow:xOrigin=312,yOrigin=162,xTarget=216,yTarget=82
Arrow:xOrigin=304,yOrigin=162,xTarget=216,yTarget=82
Arrow:xOrigin=312,yOrigin=162,xTarget=216,yTarget=80
Arrow:xOrigin=312,yOrigin=164,xTarget=216,yTarget=82

```

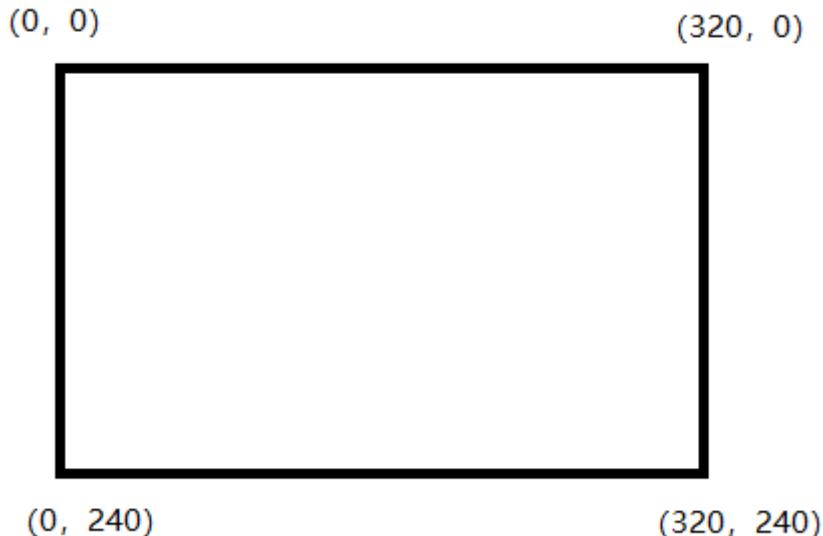
If not, you will get the results like follows:

COM4

发送

```
Block:xCenter=162,yCenter=135,width=138,height=146
Block:xCenter=163,yCenter=134,width=138,height=146
Block:xCenter=163,yCenter=134 width=138,height=146 ← width and height of the recognition frame
Block:xCenter=162,yCenter=137,width=108,height=145
Block:xCenter=162,yCenter=135,width=108,height=146
Block:xCenter=163,yCenter=135,width=138,height=146
Block:xCenter=163,yCenter=136,width=109,height=146
Block:xCenter=163,yCenter=136,width=109,height=146
Block:xCenter=162,yCenter=137,width=138,height=145
Block:xCenter=162,yCenter=136,width=138,height=146
Block:xCenter=162,yCenter=136,width=138,height=146
Block:xCenter=162,yCenter=137,width=108,height=145
Block:xCenter=162,yCenter=137 width=108,height=145 ← center coordinates of the recognition frame
Block:xCenter=163,yCenter=135,width=109,height=146
Block:xCenter=163,yCenter=137,width=109,height=145
```

Tips:The coordinate system on the HUSKYLENS screen shows as follows. It will help you to understand the coordinates in the line tracking mode.



More Documents

- Arduino Library([github](#))
- micro:bit Makecode Library([github](#))
- Mind+ Tutorial
- Protocol Document
- Tag/ QR Code Pictures
- Color Block Pictures



Get **HUSKYLENS** from DFRobot Store or **DFRobot Distributor**.

[Turn to the Top](#)

