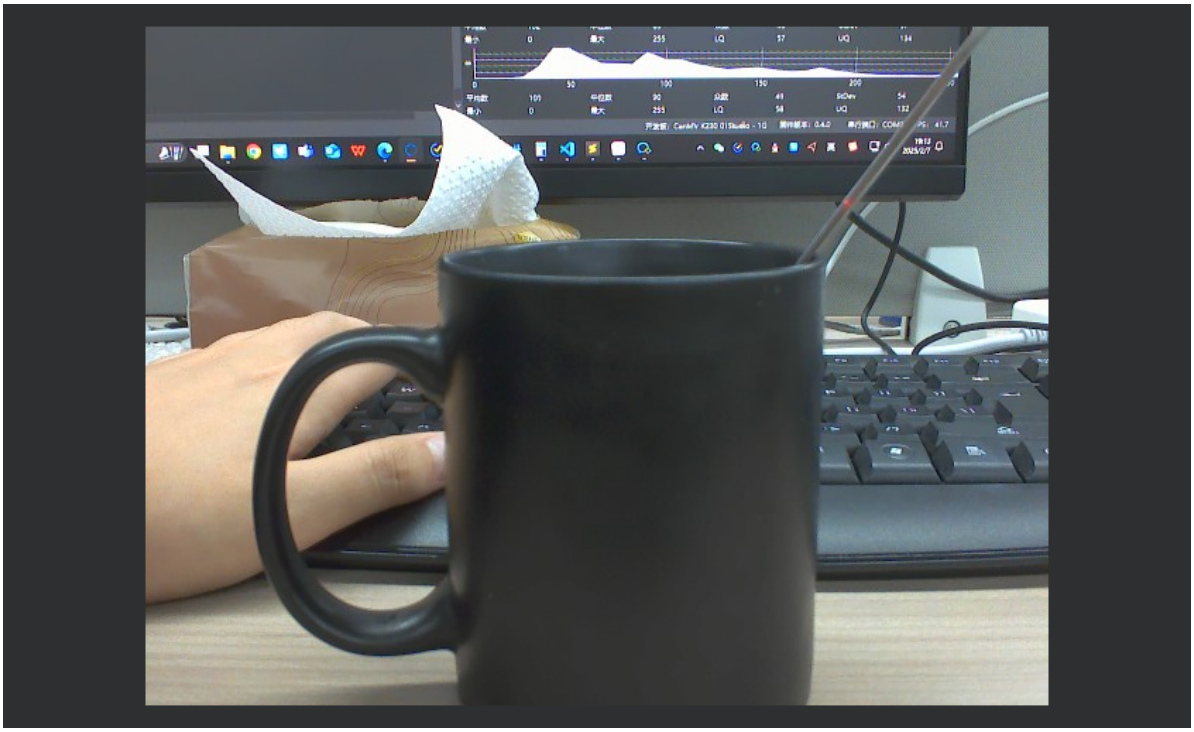
Camera image flip

The image captured by the camera before flipping is shown in the figure:



Camera horizontal mirroring (flipping)

API Description

sensor.set_hmirror

describe

Configure whether the image sensor is horizontally mirrored.

grammar

```
sensor.set_hmirror(enable)
```

parameter

Parameter name	describe	Input/Output
enable	`True` Enable horizontal mirror function `False` Disable horizontal mirror function	enter

Return Value

Return Value	describe
none	

Example

```
sensor.set_hmirror(True)
```

Routines

The code is located in [Source Code/02.Basic/18.1_sensor_hmirror.py]

```
"""
Camera preview demo
摄像头预览演示

This script initializes camera sensor, displays preview and handles cleanup
本脚本初始化摄像头传感器、显示预览并处理清理工作
"""

import sys
import uos as os
import time
from media.sensor import *
from media.display import *
from media.media import *

def init_sensor():
    """
    Initialize camera sensor with specified configuration
    使用指定配置初始化摄像头传感器
    """
    # Create sensor instance with resolution 1280x960
    # 创建分辨率为1280x960的传感器实例
    sensor = Sensor(width=1280, height=960)

    # Reset sensor to default state
    # 将传感器重置为默认状态
    sensor.reset()

    # Configure channel 1 output format to 640x480 RGB565
    # 配置通道1输出格式为640x480 RGB565
    sensor.set_framesize(width=640, height=480, chn=CAM_CHN_ID_1)
    sensor.set_pixformat(Sensor.RGB565, chn=CAM_CHN_ID_1)

    sensor.set_hmirror(True)
```

```
    return sensor

def main():
    """
    Main function to run camera preview
    运行摄像头预览的主函数
    """
    sensor = None

    try:
        # Initialize camera sensor
        # 初始化摄像头传感器
        sensor = init_sensor()

        # Initialize virtual display with 640x480 resolution
        # 初始化640x480分辨率的虚拟显示
        Display.init(Display.VIRT, width=640, height=480, to_ide=True)



        # Initialize media management
        # 初始化媒体管理
        MediaManager.init()

        # Start sensor operation
        # 启动传感器运行
        sensor.run()

        # Main loop to capture and display frames
        # 捕获和显示帧的主循环
        while True:
            # Capture frame from channel 1
            # 从通道1捕获帧
            img = sensor.snapshot(chn=CAM_CHN_ID_1)

            # Display captured frame
            # 显示捕获的帧
            Display.show_image(img)

    except KeyboardInterrupt:
        print("User interrupted the program")
        print("用户中断了程序")

    except Exception as e:
        print(f"An error occurred: {str(e)}")
        print(f"发生错误: {str(e)}")

    finally:
        # Cleanup section
        # 清理部分

        # Stop sensor if initialized
        # 如果传感器已初始化则停止
        if isinstance(sensor, Sensor):
            sensor.stop()

        # Deinitialize display
        # 反初始化显示
```

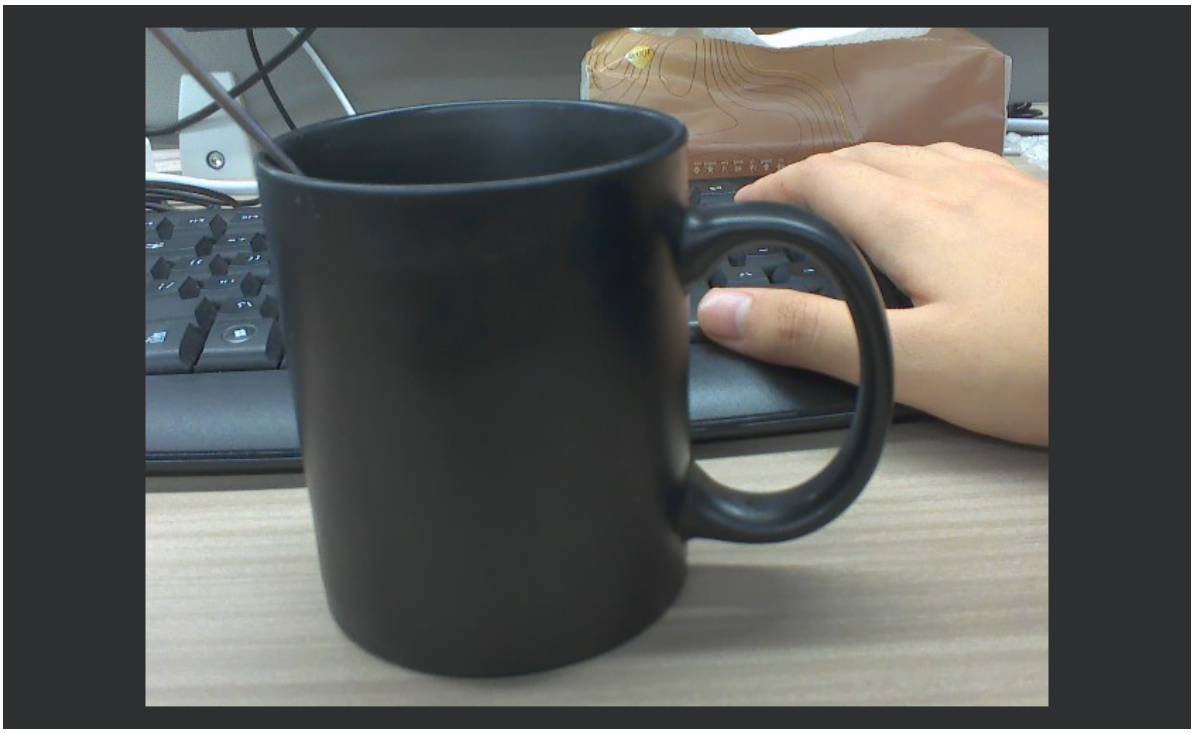
```
        Display.deinit()

        # Enable sleep mode
        # 启用睡眠模式
        os.exitpoint(os.EXITPOINT_ENABLE_SLEEP)
        time.sleep_ms(100)

        # Release media resources
        # 释放媒体资源
        MediaManager.deinit()

if __name__ == "__main__":
    main()
```

Routine execution effect:



Camera flip vertically

API Description

sensor.set_vflip

describe

Configure whether the image sensor is flipped vertically.

grammar

```
sensor.set_vflip(enable)
```

parameter

Parameter name	describe	Input/Output
enable	`True` Turn on the vertical flip function `False` Turn off the vertical flip function	enter

Return Value

Return Value	describe
none	

Example

```
sensor.set_vflip(True)
```

Routines

The code is located in [Source Code/ 02.Basic /18.2_sensor_vflip.py]

```
"""
Camera preview demo
摄像头预览演示
•
This script initializes camera sensor, displays preview and handles cleanup
本脚本初始化摄像头传感器、显示预览并处理清理工作
"""

import sys
import uos as os
import time
from media.sensor import *
from media.display import *
from media.media import *

def init_sensor():
    """
    Initialize camera sensor with specified configuration
    使用指定配置初始化摄像头传感器
    """
    # Create sensor instance with resolution 1280x960
    # 创建分辨率为1280x960的传感器实例
    sensor = Sensor(width=1280, height=960)

    # Reset sensor to default state
    # 将传感器重置为默认状态
    sensor.reset()

    # Configure channel 1 output format to 640x480 RGB565
    # 配置通道1输出格式为640x480 RGB565
    sensor.set_framesize(width=640, height=480, chn=CAM_CHN_ID_1)
    sensor.set_pixformat(Sensor.RGB565, chn=CAM_CHN_ID_1)
```

```
    sensor.set_vflip(True)

    return sensor

def main():
    """
    Main function to run camera preview
    运行摄像头预览的主函数
    """
    sensor = None

    try:
        # Initialize camera sensor
        # 初始化摄像头传感器
        sensor = init_sensor()

        # Initialize virtual display with 640x480 resolution
        # 初始化640x480分辨率的虚拟显示
        Display.init(Display.VIRT, width=640, height=480, to_ide=True)



        # Initialize media management
        # 初始化媒体管理
        MediaManager.init()

        # Start sensor operation
        # 启动传感器运行
        sensor.run()

        # Main loop to capture and display frames
        # 捕获和显示帧的主循环
        while True:
            # Capture frame from channel 1
            # 从通道1捕获帧
            img = sensor.snapshot(chn=CAM_CHN_ID_1)

            # Display captured frame
            # 显示捕获的帧
            Display.show_image(img)

    except KeyboardInterrupt:
        print("User interrupted the program")
        print("用户中断了程序")

    except Exception as e:
        print(f"An error occurred: {str(e)}")
        print(f"发生错误: {str(e)}")

    finally:
        # Cleanup section
        # 清理部分

        # Stop sensor if initialized
        # 如果传感器已初始化则停止
        if isinstance(sensor, Sensor):
            sensor.stop()
```

```
        # Deinitialize display
        # 反初始化显示
        Display.deinit()

        # Enable sleep mode
        # 启用睡眠模式
        os.exitpoint(os.EXITPOINT_ENABLE_SLEEP)
        time.sleep_ms(100)

        # Release media resources
        # 释放媒体资源
        MediaManager.deinit()

if __name__ == "__main__":
    main()
```

The execution effect of the routine is as follows: