

Lierda IC610 摄像头应用指导

产品名称：ST-A35-IC610 工业核心板

产品型号：L-IDMIM0-AA185

版本：Rev1.0

日期：25/03/15

状态：受控版本

法律声明

若接收利尔达科技集团股份有限公司(以下称为“利尔达”)的此份文档,即表示您已经同意以下条款。若不同意以下条款,请停止使用本文档。

本文档版权归利尔达科技集团股份有限公司所有,保留任何未在本文档中明示授予的权利。文档中涉及利尔达的专有信息。未经利尔达事先书面许可,任何单位和个人不得复制、传递、分发、使用和泄漏该文档以及该文档包含的任何图片、表格、数据及其他信息。

本产品符合有关环境保护和人身安全方面的设计要求,产品的存放、使用和弃置应遵照产品手册、相关合同或者相关法律、法规的要求进行。

本公司保留在不预先通知的情况下,对此手册中描述的产品进行修改和改进的权利;同时保留随时修订或收回本手册的权利。



文件修订历史

文档版本	变更日期	修订人	审核人	变更内容
Rev1.0	25-03-15	YQA		初始版本

Lierda
利 尔 达

目录

法律声明	1
文件修订历史	2
目录	3
1 引言	4
2 摄像头驱动开发	5
3 用户层操作	10



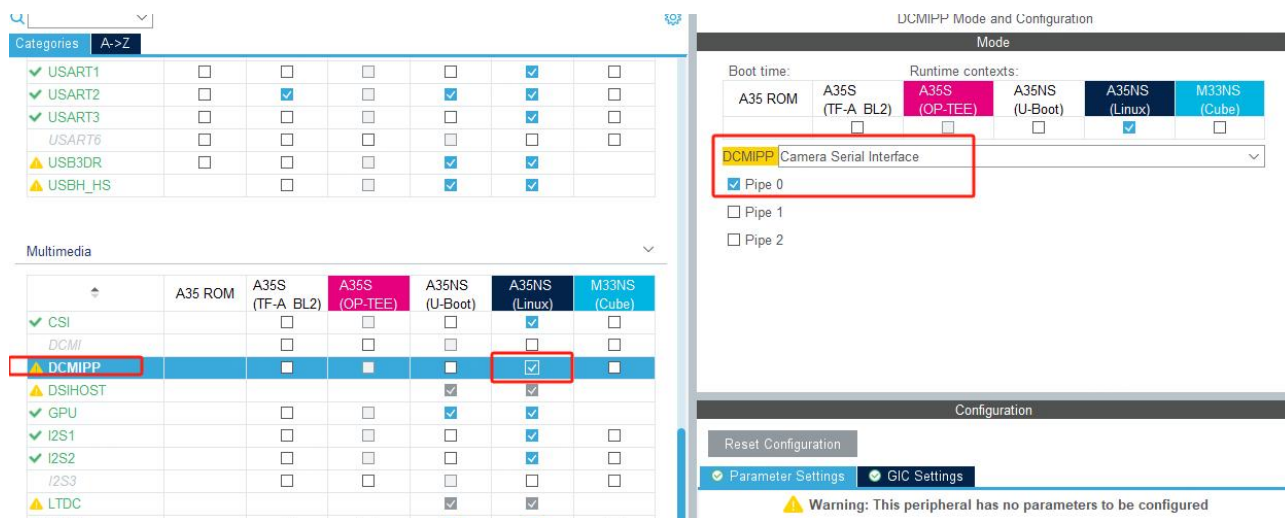
1 引言

本文档依托 IC610 evk，旨在使用 CUBEMX 构建摄像头设备树及驱动，及用户层对摄像头显示测试等。



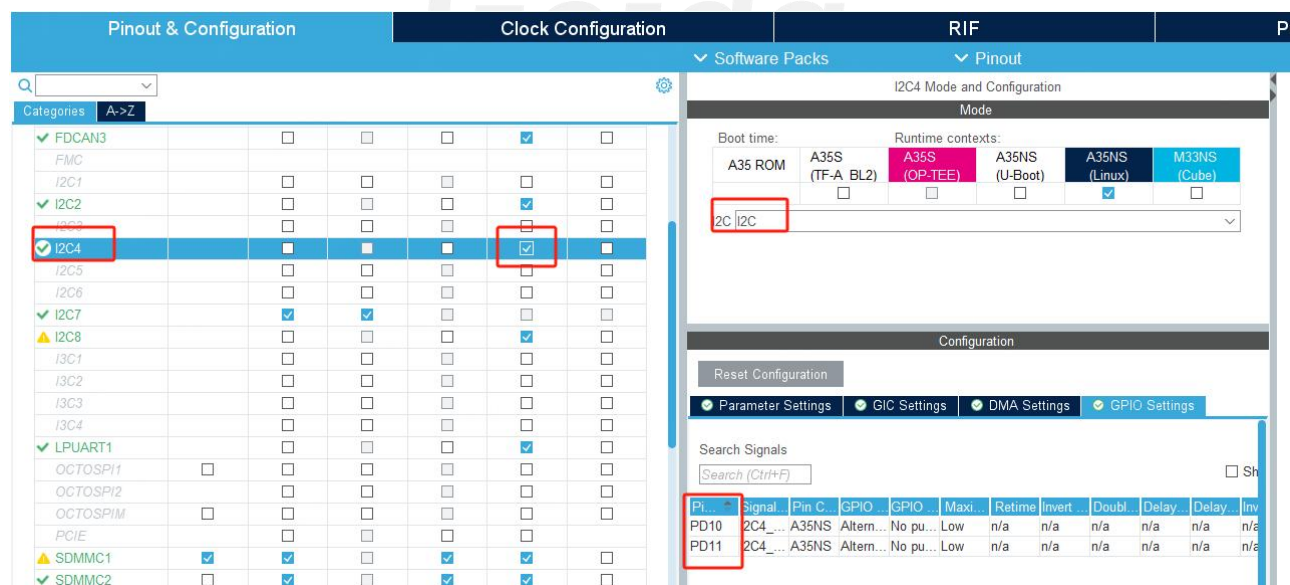
2 摄像头驱动开发

Cubemx6.14 打开 ic610.ioc，本 evk 的摄像头为 ov5645，故需要先使能 csi 总线驱动。

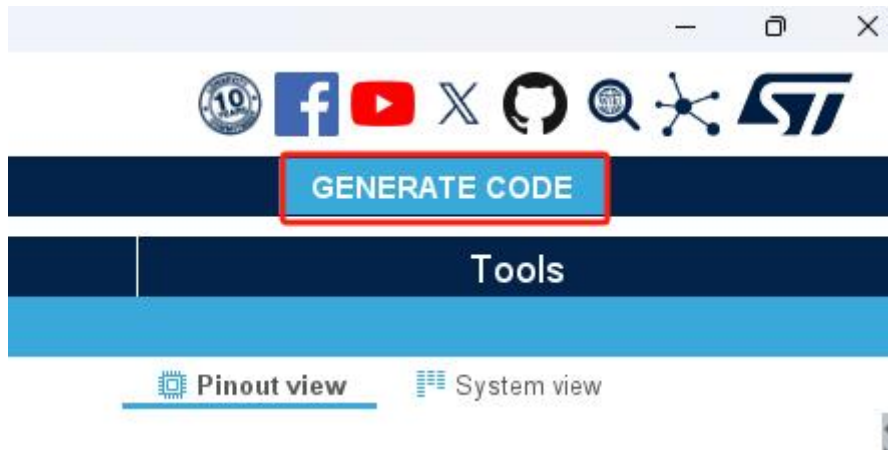


Csi 总线驱动默认使用上述配置即可，其他设置本 sdk 暂不支持。

摄像头控制引脚为 i2c，根据实际硬件设计使能 i2c 总线。如本 evk 连接摄像头的 i2c 为 i2c4，使用 gpio 为 PD10、PD11，cubemx 端使能 i2c4。



上述配置完成后点击 GENERATE CODE，即可更新 dts。



代码生成完成后，自动更新

```
ic610\CA35\DeviceTree\ic610\kernel\stm32mp255d-ic610-mx.dts
```

生成 csi 和 dcmipp 及 i2c4 节点

在以下区域内添加相关配置信息即可。

```
&csi {
    status = "okay";

    /* USER CODE BEGIN csi */
    vdd-supply = <&scmi_vddcore>;
    vdda18-supply = <&scmi_v1v8>;
    ports {
        #address-cells = <1>;
        #size-cells = <0>;
        port@0 {
            reg = <0>;
            csi_sink: endpoint {
                remote-endpoint = <&ov5645_ep>;
                data-lanes = <0 1>;
                bus-type = <4>;
            }
        }
    }
}
```

```
};  
  
};  
  
port@1 {  
    reg = <1>;  
    csi_source: endpoint {  
        remote-endpoint = <&dcmipp_0>;  
    };  
};  
  
};  
  
/* USER CODE END csi */  
  
};  
  
  
&dcmipp {  
    status = "okay";  
  
  
/* USER CODE BEGIN dcmipp */  
port {  
    dcmipp_0: endpoint {  
        remote-endpoint = <&csi_source>;  
        bus-type = <4>;  
    };  
};  
  
/* USER CODE END dcmipp */  
  
};
```



```
&i2c4 {
    pinctrl-names = "default", "sleep";
    pinctrl-0 = <&i2c4_pins_mx>;
    pinctrl-1 = <&i2c4_sleep_pins_mx>;
    status = "okay";

    /* USER CODE BEGIN i2c4 */
    ov5640: ov5640@3c {
        compatible = "ovti,ov5640";
        reg = <0x3c>;
        status = "okay";
        clocks = <&clk_ext_camera>;
        clock-names = "xclk";
        clock-frequency = <24000000>;

        DOVDD-supply = <&ov5640_vdddo_1v8>;
        AVDD-supply = <&ov5640_vdda_2v8>;
        DVDD-supply = <&ov5640_vddd_1v5>;
        rotation = <180>;

        port {
            ov5640_ep: endpoint {

                clock-lanes = <0>;
                data-lanes = <0 1>;

                remote-endpoint = <&csi_sink>;
                link-frequencies = /bits/ 64 <594000000>;
            }
        }
    }
}
```

```
};

};

};

/* USER CODE END i2c4 */
};
```

对于不同摄像头，执行修改 csi 下 remote-endpoint = <&ov5640_ep>; 即可。

同时 kernel 下使能摄像头驱动 CONFIG_VIDEO_OV5640，重新编译全部镜像并更新到开发板中。

Lierda
利 尔 达

3 用户层操作

摄像头预览,可实时输出摄像头内容到屏幕:

```
/usr/local/demo/application/camera/bin/launch_camera_preview_mp25.sh
```

添加其他摄像头可修改如下脚本:

```
/usr/local/demo/application/camera/bin/launch_camera_control_mp25.sh
```

如下增加 ov5645

```
case "$sensor_subdev" in
    *"ov5640"*)
        #OV5640 only support 720p with raw-bayer format
        SENSORWIDTH=1280
        SENSORHEIGHT=720
        #OV5640 claims to support all raw bayer combinations but always output
        SBGGR8_1X8...
        sensorbuscode=SBGGR8_1X8
        ;;
    *"ov5645"*)
        SENSORWIDTH=1280
        SENSORHEIGHT=720
        sensorbuscode=YUYV8_2X8
    *"imx335"*)
        #IMX335 expose both RGGB10 and RGGB12 however only RGGB10
        can work on CSI 2 lanes
        sensorbuscode=SRGGB10_1X10
        main_postproc=$(media-ctl -d $DCMIPP_MEDIA -e
        dcmipp_main_postproc)
```

```

#Enable gamma correction
v4l2-ctl -d "$main_postproc" -c gamma_correction=1

#Do exposure correction continuously in background
sleep 3  && while : ; do /usr/local/demo/bin/dcmipp-isp-ctrl -i0 -g >
/dev/null ; done &

;;

main_postproc=$(media-ctl          -d          $DCMIPP_MEDIA          -e
dcmipp_main_postproc)

#Enable gamma correction
v4l2-ctl -d "$main_postproc" -c gamma_correction=1

#Do exposure correction continuously in background
sleep 3  && while : ; do /usr/local/demo/bin/dcmipp-isp-ctrl -i0 -g >
/dev/null ; done &

;;

```