ochin_CM4v2 Hardware test number 15

KTD2025 RGB LED test

Devices used for tests

1. ochin CM4v2 carrier board
2. Raspberry Pi CM4 module with eMMC
3. Power Supply 0-30Vdc

Test description

The purpose of this test is to verify the proper functioning of the RGB LED located on the external board ochin board.

Management of the RGB LED is entrusted to a chip called KTD2026, which is connected to the CM4 module via the I2C1 interface.

Using the I2C libraries, it is possible to configure the chip to vary the intensity of each of the three basic colours and their respective transitions. For more precise and detailed information on the handling of the chip, it is necessary to study the datasheet of the KTD2026, available on the website of the manufacturer, Kinetic Technologies (https://www.kinet-ic.com).

Preliminary configuration

In order to test the INA219 chip, it is necessary to enable the I2C1 interface.
The I2C interface could be enabled via raspi-config or directly adding the following lines in the boot/config.txt file:

```
dtparam=i2c_arm=on,i2c_arm_baudrate=400000
```

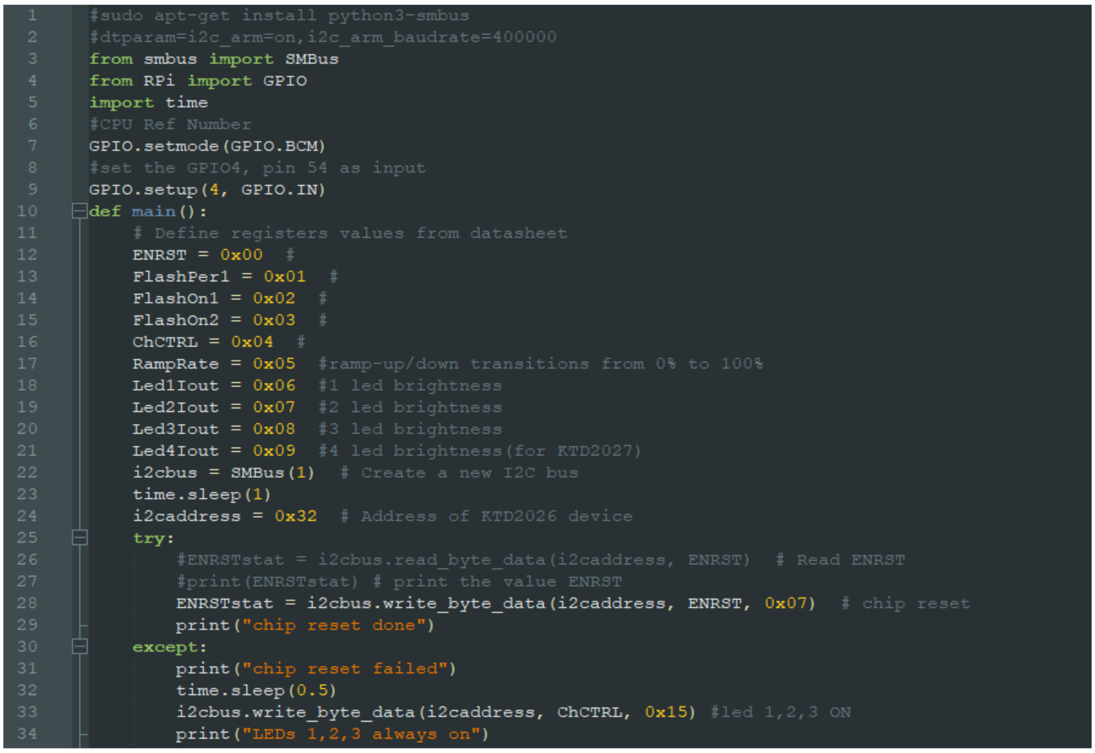
To access the I2C interface via Python, the libraries must first be installed:

```
sudo apt-get install python3-smbus
```

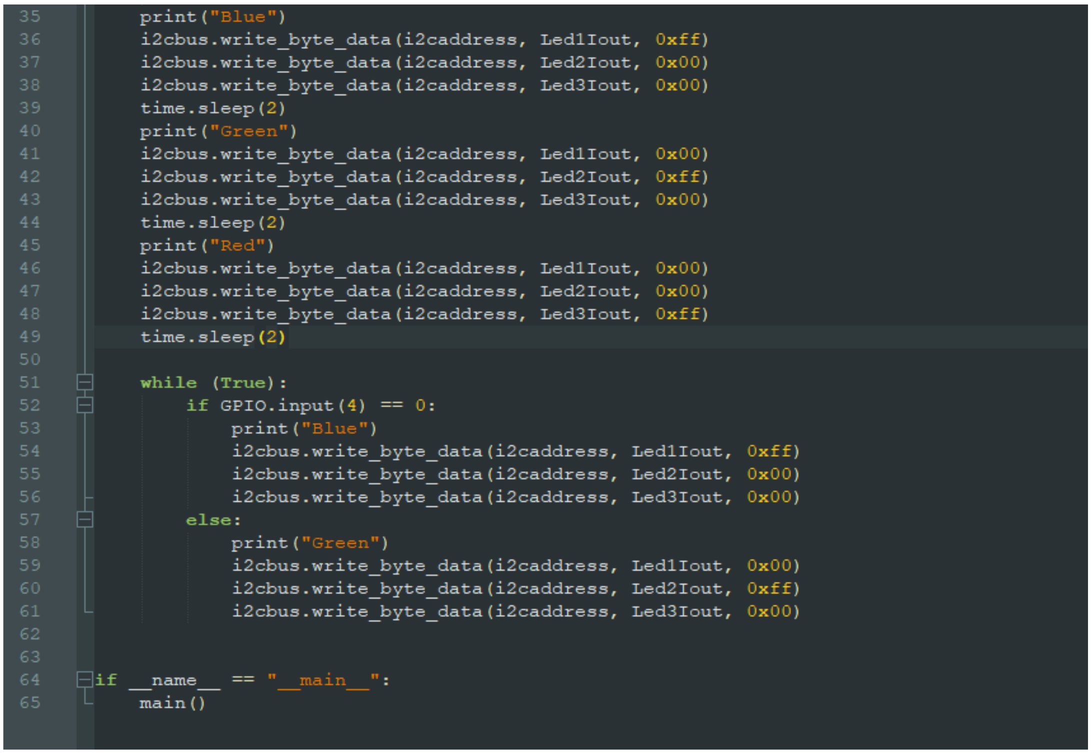
Test execution

In order to test the S1 button and the KTD2026, simply run the python script " KTD2026_test" :

```
python KTD2026_test
```



```python
#sudo apt-get install python3-smbus
#dtparam=i2c_arm=on,i2c_arm_baudrate=400000
from smbus import SMBus
from RPi import GPIO
import time
#CPU Ref Number
GPIO.setmode(GPIO.BCM)
#set the GPIO4, pin 54 as input
GPIO.setup(4, GPIO.IN)
def main():
    # Define registers values from datasheet
    ENRST = 0x00  #
    FlashPer1 = 0x01  #
    FlashOn1 = 0x02  #
    FlashOn2 = 0x03  #
    ChCTRL = 0x04  #
    RampRate = 0x05  #ramp-up/down transitions from 0% to 100%
    Led1Iout = 0x06  #1 led brightness
    Led2Iout = 0x07  #2 led brightness
    Led3Iout = 0x08  #3 led brightness
    Led4Iout = 0x09  #4 led brightness(for KTD2027)
    i2cbus = SMBus(1)  # Create a new I2C bus
    time.sleep(1)
    i2caddress = 0x32  # Address of KTD2026 device
    try:
        #ENRSTstat = i2cbus.read_byte_data(i2caddress, ENRST)  # Read ENRST
        #print(ENRSTstat) # print the value ENRST
        ENRSTstat = i2cbus.write_byte_data(i2caddress, ENRST, 0x07)  # chip reset
        print("chip reset done")
    except:
        print("chip reset failed")
        time.sleep(0.5)
        i2cbus.write_byte_data(i2caddress, ChCTRL, 0x15) #led 1,2,3 ON
        print("LEDs 1,2,3 always on")
```



```
    print("Blue")
    i2cbus.write_byte_data(i2caddress, Led1Iout, 0xff)
    i2cbus.write_byte_data(i2caddress, Led2Iout, 0x00)
    i2cbus.write_byte_data(i2caddress, Led3Iout, 0x00)
    time.sleep(2)
    print("Green")
    i2cbus.write_byte_data(i2caddress, Led1Iout, 0x00)
    i2cbus.write_byte_data(i2caddress, Led2Iout, 0xff)
    i2cbus.write_byte_data(i2caddress, Led3Iout, 0x00)
    time.sleep(2)
    print("Red")
    i2cbus.write_byte_data(i2caddress, Led1Iout, 0x00)
    i2cbus.write_byte_data(i2caddress, Led2Iout, 0x00)
    i2cbus.write_byte_data(i2caddress, Led3Iout, 0xff)
    time.sleep(2)

    while (True):
        if GPIO.input(4) == 0:
            print("Blue")
            i2cbus.write_byte_data(i2caddress, Led1Iout, 0xff)
            i2cbus.write_byte_data(i2caddress, Led2Iout, 0x00)
            i2cbus.write_byte_data(i2caddress, Led3Iout, 0x00)
        else:
            print("Green")
            i2cbus.write_byte_data(i2caddress, Led1Iout, 0x00)
            i2cbus.write_byte_data(i2caddress, Led2Iout, 0xff)
            i2cbus.write_byte_data(i2caddress, Led3Iout, 0x00)


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

If the driver is working properly, at first the LED will be colored the three basic colors (Blue, Green, Red) and then all combinations will be displayed in an infinite cycle.

Test result

Test passed