|  |
| --- |
|  |
| Arduino Core for Duo测试报告 |
|  |

目录

[测试报告 3](#__RefHeading___Toc4104_1285584725)

[测试环境 3](#__RefHeading___Toc4106_1285584725)

[基本接口测试 4](#__RefHeading___Toc4108_1285584725)

[Digital I/O测试 5](#__RefHeading___Toc4110_1285584725)

[支持接口 5](#__RefHeading___Toc4112_1285584725)

[测试案例 5](#__RefHeading___Toc4114_1285584725)

[Analog I/O测试 8](#__RefHeading___Toc4116_1285584725)

[支持接口 8](#__RefHeading___Toc4118_1285584725)

[测试案例 8](#__RefHeading___Toc4120_1285584725)

[Advance I/O测试 11](#__RefHeading___Toc4122_1285584725)

[支持接口 11](#__RefHeading___Toc4124_1285584725)

[测试案例 12](#__RefHeading___Toc4126_1285584725)

[Time测试 15](#__RefHeading___Toc4128_1285584725)

[支持接口 15](#__RefHeading___Toc4130_1285584725)

[测试案例 16](#__RefHeading___Toc4132_1285584725)

[External Interrupt测试 17](#__RefHeading___Toc4134_1285584725)

[支持接口 17](#__RefHeading___Toc4136_1285584725)

[测试案例 17](#__RefHeading___Toc4138_1285584725)

[Serial 测试 18](#__RefHeading___Toc4140_1285584725)

[支持接口 18](#__RefHeading___Toc4142_1285584725)

[测试案例 18](#__RefHeading___Toc4144_1285584725)

[SPI测试 22](#__RefHeading___Toc4146_1285584725)

[支持接口 22](#__RefHeading___Toc4148_1285584725)

[测试案例 22](#__RefHeading___Toc4150_1285584725)

[Wire 测试 25](#__RefHeading___Toc4152_1285584725)

[支持接口 25](#__RefHeading___Toc4154_1285584725)

[测试案例 25](#__RefHeading___Toc4156_1285584725)

[MailBox测试 27](#__RefHeading___Toc4158_1285584725)

[其它测试 31](#__RefHeading___Toc4160_1285584725)

[支持接口 31](#__RefHeading___Toc4162_1285584725)

[测试案例 32](#__RefHeading___Toc4164_1285584725)

[整合测试 37](#__RefHeading___Toc4166_1285584725)

[超声测距测试 37](#__RefHeading___Toc4168_1285584725)

[场景描述 37](#__RefHeading___Toc4170_1285584725)

[测试程序 38](#__RefHeading___Toc4172_1285584725)

[伺服马达驱动测试 40](#__RefHeading___Toc4174_1285584725)

[场景描述 40](#__RefHeading___Toc4176_1285584725)

[测试程序 41](#__RefHeading___Toc4178_1285584725)

[IIC SSD1306测试 42](#__RefHeading___Toc4180_1285584725)

[场景描述 42](#__RefHeading___Toc4182_1285584725)

[测试程序 43](#__RefHeading___Toc4184_1285584725)

[SPI SD文件系统测试 45](#__RefHeading___Toc4186_1285584725)

[场景描述 45](#__RefHeading___Toc4188_1285584725)

[测试程序 47](#__RefHeading___Toc4190_1285584725)

[LVGL测试 51](#__RefHeading___Toc4192_1285584725)

[场景描述 51](#__RefHeading___Toc4194_1285584725)

[测试程序 53](#__RefHeading___Toc4196_1285584725)

测试报告

本报告将对Arduino Core for Duo的接口测试以及整合测试进行详细就介绍。

# 测试环境

本文档基于milk-v duo和milk-v duo256进行测试，使用Arduino IDE进行代码编写和测试，使用IDE自带示例代码和编写的测试用例进行测试

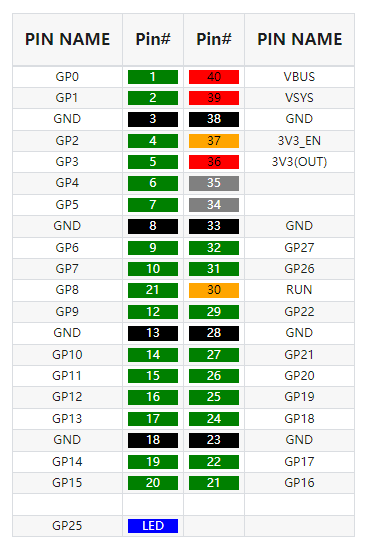
**硬件环境**

|  |  |
| --- | --- |
| Milk-V Duo | 规格 |
| 处理器 | CVITEK CV1800B (C906@1Ghz + C906@700MHz) |
| RAM | DDR2 64MB/256MB/(512MB) |
| 存储器 | 1x Mirco SD 插槽,1x SD NAND 焊盘 |
| USB | 1x Type-C 用于数据和电源，1x USB2 焊盘 |
| Camera | 1x 16P FPC 连接器（MIPI CSI 2 通道） |
| 通用输入输出口 | 多达 26 个 Pins 可用于通用 I/O（GPIO） |
| 尺寸 | 21mm \* 51mm |
| 外设 | Up to 26 GPIO pins on the MilkV-Duo 40-pin header provide access to internal peripherals such as SDIO, I2C, PWM, SPI, J-TAG, and UART.  Up to 3x I2C  Up to 5x UART  Up to 1x SDIO1  Up to 1x SPI  Up to 2x ADC  Up to 7x PWM  Up to 1x RUN  Up to 1x JTAG  集成 MAC PHY 支持 10/100Mbps 全双工或半双工模式  一个 USB Host / device 接口 |

**软件环境**

|  |  |
| --- | --- |
| 软件 | 下载地址 |
| Arduino IDE(2.2.1) | <https://www.arduino.cc/en/software> |

Pin 示例图



# 基本接口测试

该类测试主要针对Arduino Core for Duo的接口进行测试。根据官方的语言描述，Arduino主要提供Digital I/O, Analog I/O, Advance I/O等类别的接口，接下来将分别对它们进行测试。

## Digital I/O测试

### 支持接口

void pinMode(uint8\_t pinNumber, uint8\_t pinMode)

int digitalRead(uint8\_t pinNumber)

void digitalWrite(uint8\_t pinNumber, uint8\_t status)

### 测试案例

#### 案例1：LED测试

LED灯两端连接TEST\_PIN和GND,

示例以PIN 20为例

LED长管脚接PIN20

LED短管脚接GND也就是PIN18

#define TEST\_PIN 20 //0,1,2,14,15,19,20,21,22,24,25,26,27

// the setup function runs once when you press reset or power the board

void setup() {

pinMode(TEST\_PIN, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(TEST\_PIN, HIGH); // turn the LED on (HIGH is the voltage level)

delay(500); // wait for a second

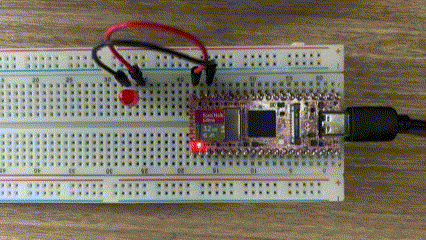
digitalWrite(TEST\_PIN, LOW); // turn the LED off by making the voltage LOW

delay(500); // wait for a second

}

**测试结果：**

LED灯会闪烁



#### 案例2：GPIO读取

连接PIN 19 和 TEST\_PIN， LED灯两端连接PIN1和GND

#define TEST\_PIN 20 //2,14,15,20,21,22,24,25,26,27

void setup() {

// initialize digital pin LED\_BUILTIN as an output

pinMode(LED\_BUILTIN, OUTPUT);

pinMode(TEST\_PIN, INPUT);

pinMode(19, OUTPUT); // 19 <-> TEST\_PIN ==> 1

pinMode(1, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(LED\_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)

digitalWrite(19, HIGH); // turn the LED on (HIGH is the voltage level)

delay(50);

int data = digitalRead(TEST\_PIN) == HIGH ? LOW : HIGH;

digitalWrite(1, data); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW

digitalWrite(19, LOW); // turn the LED on (HIGH is the voltage level)

delay(50);

data = digitalRead(TEST\_PIN) == HIGH ? LOW : HIGH;

digitalWrite(1, data); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

}

**测试结果：**

PIN1 LED灯会和Duo开发板自带蓝灯交替闪烁。

## Analog I/O测试

### 支持接口

void analogWriteResolution(int bits)

void analogReadResolution(int bits)

void analogWrite(uint8\_t pinNumber, uint32\_t val)

uint32\_t analogRead(uint32\_t pinNumber)

void analogReference(uint8\_t mode)

其中analogReference并无实际功能，analogReadResolution配合analogRead起作用，类似的analogWriteResolution配合analogWrite起作用。

### 测试案例

#### 案例1：ADC读取

传感器数据输出线连接PIN31

UART使用PIN6和PIN7

int analogPin = PIN\_ADC1;

int val = 0;

void setup() {

Serial2.begin(115200); // setup serial

}

void loop() {

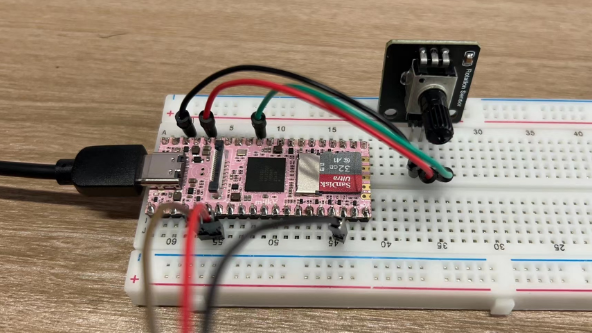
val = analogRead(analogPin); // read the input pin

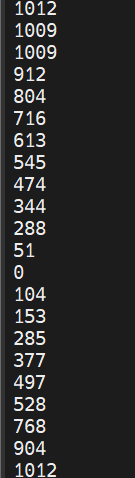
Serial2.println(val); // debug value

}

**测试结果：**

输出读取旋转编码器数值





#### 案例2：输出PWM波形

void setup() {

analogWriteResolution(14); // 20ms

Serial.begin(115200);

analogWrite(4, 500);

analogWrite(5, 1000);

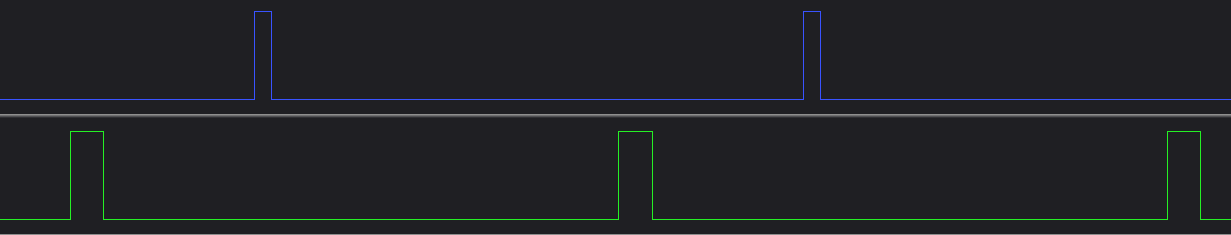
}

void loop() {

}

**测试结果：**

PIN4和PIN5（连接逻辑分析仪）输出波形：



## Advance I/O测试

### 支持接口

void tone(pin\_size\_t pinNumber, unsigned int frequency)

void tone(pin\_size\_t pinNumber, unsigned int frequency, unsigned long duration)

void noTone(pin\_size\_t pin)

uint8\_t shiftIn(pin\_size\_t ulDataPin, pin\_size\_t ulClockPin, BitOrder ulBitOrder)

void shiftOut(pin\_size\_t ulDataPin, pin\_size\_t ulClockPin, BitOrder ulBitOrder, uint8\_t ulVal)

unsigned long pulseIn(uint8\_t pin, uint8\_t state, unsigned long timeout)

unsigned long pulseInLong(uint8\_t pin, uint8\_t state, unsigned long timeout)

### 测试案例

#### 案例1：tone/noTone

void setup() {

tone(4, 1000);

}

void loop() {

tone(5, 1000);

delay(30);

noTone(4);

tone(5, 2000);

delay(30);

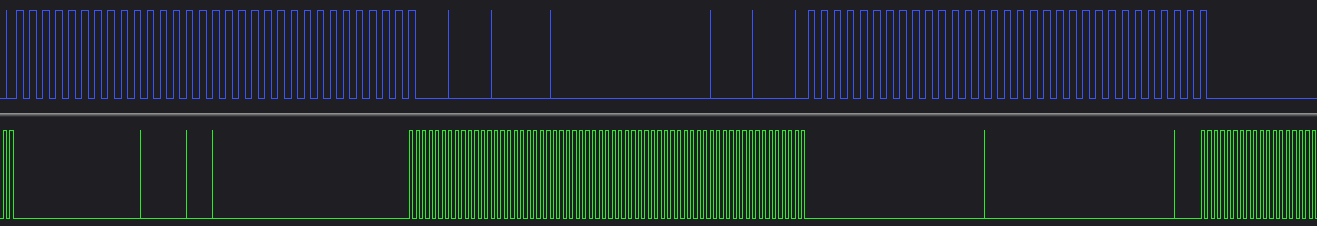
noTone(5);

tone(4, 1000);

}

**测试结果：**

PIN4和PIN5（连接逻辑分析仪）交替输出方波（1k HZ 和 2k HZ）：



#### 案例2：shiftIn/shiftOut

连接PIN2 和PIN4 （在Duo256上测试需要用最新Arduino-sophgo代码）

const int outDataPin = 21;

const int clkPin = 20;

const int inDataPin = 2;

void setup() {

pinMode(outDataPin, OUTPUT);

pinMode(clkPin, OUTPUT);

pinMode(inDataPin, INPUT);

tone(4, 1000); // connect 4 to inDataPin

Serial.begin(115200);

}

void loop() {

shiftOut(outDataPin, clkPin, MSBFIRST, 0x5A);

delay(100);

byte data = shiftIn(inDataPin, clkPin, MSBFIRST);

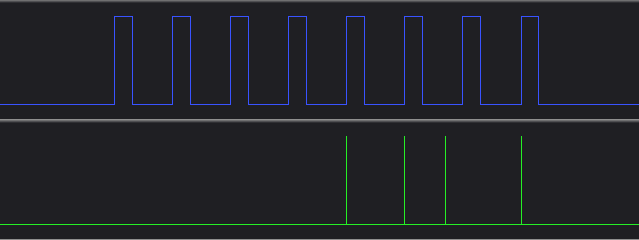
Serial.printf("shift in %x\r\n", data);

delay(100);

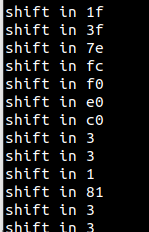
}

**测试结果：（无外联设备，连接逻辑分析仪）**

波形波形如下：



串口输出如下



#### 案例3：pulseIn/pulseInLong

连接PIN2 和PIN4 （在Duo256上测试需要用最新Arduino-sophgo代码）

const int pulsePin = 2;

void setup() {

pinMode(pulsePin, INPUT);

Serial.begin(115200);

tone(4, 1000); // connect 4 to pulsePin

}

void loop() {

unsigned long duration = pulseInLong(pulsePin, HIGH, 1000000);

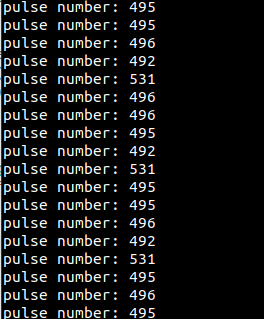
Serial.printf("pulse number: %ld\r\n", duration);

delay(1000);

}

**测试结果：**

串口输出如下：



## Time测试

### 支持接口

unsigned long millis(void);

unsigned long micros(void);

void delay(unsigned long);

void delayMicroseconds(unsigned long us);

### 测试案例

void setup() {

Serial.begin(115200);

}

void loop() {

unsigned long milli = millis();

delay(1000);

Serial.printf("time: %ld ms\r\n", millis() - milli);

unsigned long micro = micros();

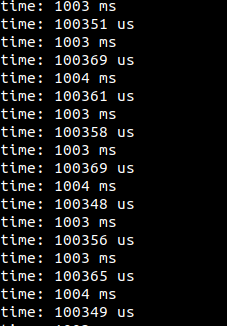
delayMicroseconds(100000);

Serial.printf("time: %ld us\r\n", micros() - micro);

}

**测试结果：**

串口输出如下：



## External Interrupt测试

### 支持接口

void attachInterrupt(pin\_size\_t pin, void (\*callback)(void), uint32\_t mode);

void detachInterrupt(pin\_size\_t pin);

### 测试案例

连接PIN2 和PIN1

const byte ledPin = 1;

const byte interruptPin = 2;

volatile byte state = HIGH;

void setup() {

Serial.begin(115200);

pinMode(ledPin, OUTPUT);

pinMode(interruptPin, INPUT\_PULLUP);

digitalWrite(ledPin, LOW);

attachInterrupt(digitalPinToInterrupt(interruptPin), blink, RISING);

}

void loop() {

state = !state;

digitalWrite(ledPin, state);

delay(1000);

}

void blink() {

Serial.printf("trigger interrupt\n\r");

digitalWrite(ledPin, HIGH);

detachInterrupt(digitalPinToInterrupt(interruptPin));

}

**测试结果：**

串口输出一次 trigger interrupt：

descript

## Serial 测试

### 支持接口

void begin(unsigned long baud, uint32\_t config=SERIAL\_8N1, int8\_t rxPin=-1, int8\_t txPin=-1);

size\_t read(uint8\_t \*buffer, size\_t size);

size\_t write(const uint8\_t \*buffer, size\_t size);

operator bool() const;...

...

还有其它继承自Stream/Print的方法，如print/printf等

### 测试案例

#### 案例1：普通输出

void setup() {

Serial.begin(115200);

}

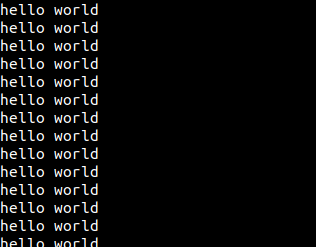
void loop() {

Serial.printf("hello world\r\n");

}

**测试结果：**

串口不断输出hello world：



#### 案例2：多Serial交互

void setup() {

Serial2.begin(115200);

}

bool a = true;

void loop() {

//read from port 2, send to port 3:

if (a && Serial2.available()) {

int inByte = Serial2.read();

Serial3.begin(115200);

Serial3.printf("From 2: receive %c\n\r", inByte);

int i = 0;

while(Serial2.available()) {

inByte = Serial2.read();

Serial3.printf(" -- %d: ", i++);

Serial3.write(inByte);

}

Serial3.printf("2 Msg end\n\r");

a = false;

}

//read from port 3, send to port 2:

if (!a && Serial3.available()) {

int inByte = Serial3.read();

Serial2.begin(115200);

Serial2.printf("From 3: receive %c\n\r", inByte);

int i = 0;

while(Serial3.available()) {

inByte = Serial3.read();

Serial2.printf(" -- %d: ", i++);

Serial2.write(inByte);

}

Serial2.printf("3 Msg end\n\r");

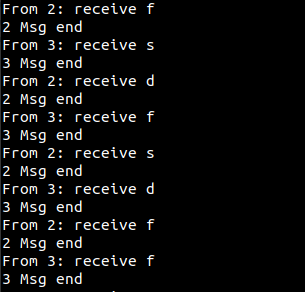
a = true;

}

}

**测试结果：**

串口2/3 交替传递单字符：



## SPI测试

### 支持接口

void begin(int8\_t sckPin, int8\_t misoPin, int8\_t mosiPin, int8\_t csPin);

void beginTransaction(SPISettings settings);

void endTransaction(void);

void transfer(void \* data, uint32\_t size);

uint8\_t transfer(uint8\_t data);

uint16\_t transfer16(uint16\_t data);

...

更多介绍见<https://www.arduino.cc/reference/en/language/functions/communication/spi/>

### 测试案例

案例1： spi loopback

连接PIN10和PIN11

#include <SPI.h>

char str[]="hello world\n";

void setup() {

// put your setup code here, to run once:

Serial2.begin(115200);

SPI.begin();

}

byte i = 0;

void loop() {

// put your main code here, to run repeatedly:

// digitalWrite(12, 1);

SPI.beginTransaction(SPISettings());

Serial2.printf("transfer %c\n\r", str[i]);

char out = SPI.transfer(str[i++]); // spi loop back

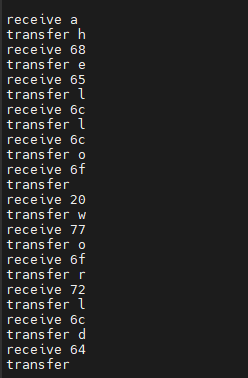
SPI.endTransaction();

Serial2.printf("receive %x \n\r", out);

i %= 12;

}

测试结果



## Wire 测试

### 支持接口

void begin(csi\_iic\_addr\_mode\_t addr\_mode = IIC\_ADDRESS\_7BIT);

void begin(uint16\_t address, csi\_iic\_addr\_mode\_t addr\_mode = IIC\_ADDRESS\_7BIT);

void beginTransmission(uint16\_t);

uint8\_t endTransmission(bool stopBit);

uint8\_t endTransmission(void);

size\_t requestFrom(uint16\_t address, size\_t quantity, bool stopBit);

size\_t requestFrom(uint16\_t address, size\_t quantity);

size\_t write(uint8\_t data);

size\_t write(const uint8\_t \* data, size\_t quantity);

virtual int available(void);

virtual int read(void);

void onReceive(void(\*)(int));

void onRequest(void(\*)(void));

...

更多介绍见<https://www.arduino.cc/reference/en/language/functions/communication/wire/>

### 测试案例

#### 案例1：IIC0向IIC1发送数据

引脚连接：连接IIC0和IIC1对应的接口，对应关系如下：

|  |  |
| --- | --- |
| IIC0 | IIC1 |
| 1 | 11 |
| 2 | 12 |

IIC0（master） 和 IIC1（slave）

下面测试仅适用于Duo（Duo256没有IIC0对外接口，可以根据需要改成IIC1（Wire1）和IIC2（Wire2）的通信）

#include <Wire.h>

void receive(int a) {

Serial.printf("receive %d bytes\n\r", a);

while(a--) {

Serial.printf("%d \n\r", Wire1.read());

}

}

void setup() {

Serial.begin(115200);

Wire1.begin(0x50);

Wire1.onReceive(receive);

Wire.begin();

Serial.printf("test slave\n\r");

Wire1.print();

}

byte val = 0;

void loop() {

// put your main code here, to run repeatedly:

Wire.beginTransmission(0x50); // Transmit to device number 0x50

Serial.printf("send %d \n\r", ++val);

Wire.write(val); // Sends value byte

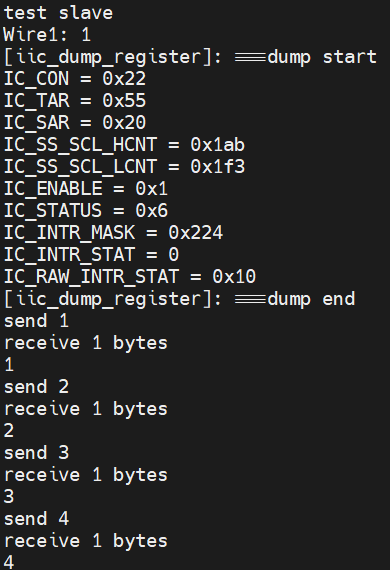
Wire.endTransmission(); // Stop transmitting

Wire1.onService();

delay(1000);

}

**测试结果：**



## MailBox测试

Duo目前使用rtos\_cmdqu驱动实现大小核通信详见：<https://milkv.io/docs/duo/getting-started/rtoscore>

大核测试程序

见mailbox\_test.c

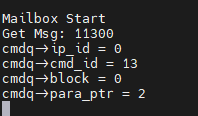
ino测试程序

见mailbox.ino

测试结果大核运行

descript

小核收到



其中

cmd\_id为大核程序定义值，CMD\_DUO\_LED=0x13  
para\_ptr为大核定义命令值 DUO\_LED\_ON =0x2

## 其它测试

### 支持接口

主要包含Math， Trigonometry，Characters ，Random Numbers以及Bits and Bytes

等接口，见<https://www.arduino.cc/reference/en/>

### 测试案例

#### 案例1：Math/Random Numbers

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);

}

void loop() {

int num = random(0, 100);

Serial.printf("abs %d = %d\r\n", -num, abs(-num));

Serial.printf("abs %d = %d\r\n", num, abs(num));

Serial.printf("constrain %d 30 60 = %d\r\n", num, constrain(num, 30, 50));

Serial.printf("map %d 0, 100, -100, 100 = %d\r\n", num, map(num, 0, 100, -100, 100));

Serial.printf("map %d 0, 100, 100, -100 = %d\r\n", num, map(num, 0, 100, 100, -100));

Serial.printf("max %d 50 = %d\r\n", num, max(num, 50));

Serial.printf("min %d 50 = %d\r\n", num, min(num, 50));

Serial.printf("pow %d 2 = %d\r\n", num, (int)pow(num, 2));

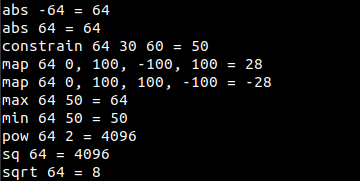
Serial.printf("sq %d = %d\r\n", num, (int)sq(num));

Serial.printf("sqrt %d = %d\r\n", num, (int)sqrt(num));

}

**测试结果：**

串口输出如下：



#### 案例2：Trigonometry

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);

Serial.printf("trig PI/6 result(x1000) => cos: %d\t sin:%d\t tan: %d\r\n", (int)(cos(PI/6) \* 1000), (int)(sin(PI/6) \* 1000), (int)(tan(PI/6)\* 1000));

Serial.printf("trig PI/3 result(x1000) => cos: %d\t sin:%d\t tan: %d\r\n", (int)(cos(PI/3) \* 1000), (int)(sin(PI/3) \* 1000), (int)(tan(PI/3)\* 1000));

Serial.printf("trig PI/2 result(x1000) => cos: %d\t sin:%d\t tan: %d\r\n", (int)(cos(PI/2) \* 1000), (int)(sin(PI/2) \* 1000), (int)(tan(PI/2)\* 1000));

Serial.printf("trig PI\*2/3 result(x1000) => cos: %d\t sin:%d\t tan: %d\r\n", (int)(cos(PI\*2/3) \* 1000), (int)(sin(PI\*2/3) \* 1000), (int)(tan(PI\*2/3)\* 1000));

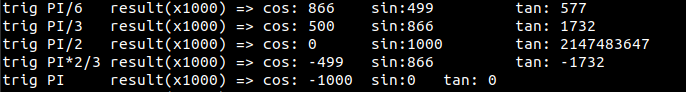
Serial.printf("trig PI result(x1000) => cos: %d\t sin:%d\t tan: %d\r\n", (int)(cos(PI) \* 1000),(int)(sin(PI) \* 1000), (int)(tan(PI)\* 1000));

}

void loop() {}

**测试结果：**

串口输出如下：

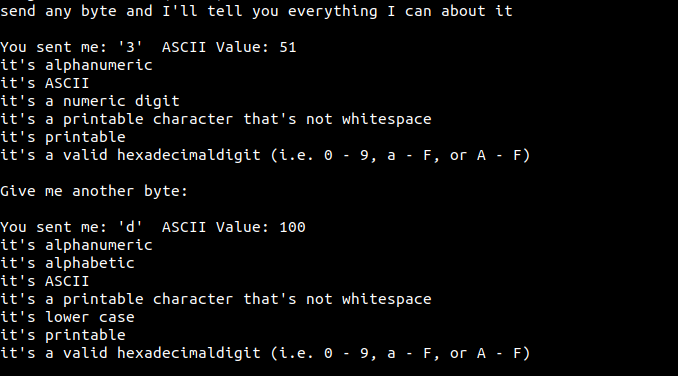


#### 案例3：Characters

见character.ino文件

**测试结果：**

串口输出如下：



#### 案例4：Bits and Bytes

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);

}

void loop() {

long value = random(0, 0xFFFFFFFF);

long num = random(0, 31);

Serial.printf("bit %d = %x\r\n", num, bit(num));

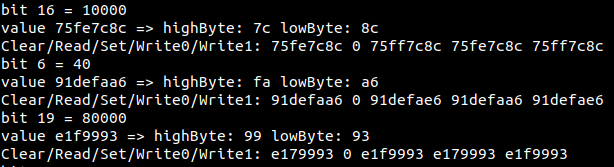
Serial.printf("value %x => highByte: %x lowByte: %x\r\n", value, highByte(value), lowByte(value));

Serial.printf("Clear/Read/Set/Write0/Write1: %x %x %x %x %x \r\n", bitClear(value, num), bitRead(value, num), bitSet(value, num), bitWrite(value, num, 0), bitWrite(value, num, 1));

}

**测试结果：**

串口输出如下：



# 整合测试

该类测试主要是针对特定应用场景对当前Arduino Core for Duo 进行测试，例如超声测据，马达驱动，LCD显示，LVGL库支持等。

## 超声测距测试

### 场景描述

通过HC-SR04进行超声测距（3.3V电源）。

* TrigPin连接PIN19
* EchoPin链接PIN20



### 测试程序

/\*

\* HC-SR04 example sketch

\*

\* https://create.arduino.cc/projecthub/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-036380

\*

\* by Isaac100

\*/

const int trigPin = 19;

const int echoPin = 20;

volatile byte state = HIGH;

float duration, distance;

void setup() {

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

Serial.begin(115200);

}

void loop() {

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration\*.0343)/2;

Serial.print("Distance: ");

Serial.println(distance);

delay(100);

}

**测试结果：**

随着遮挡物移动输出距离也跟着变化，如下所示：



## 伺服马达驱动测试

### 场景描述

通过PWM接口驱动sg90马达进行转动，需要采用5V电源

* 信号线(橘色) <=> PIN4
* VCC(红色) <=> 5V电源
* GND(褐色) <=> GND



### 测试程序

void setup() {

analogWriteResolution(14); // 20ms

Serial.begin(115200);

}

int i = 0;

void loop() {

int duty = 500 + i \* 500;

analogWrite(4, duty);

Serial.printf("become %d\n\r", duty);

delay(1000);

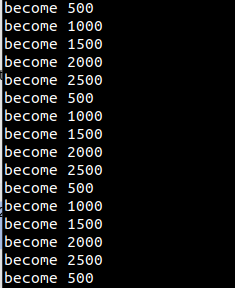
i++;

i %= 5;

}

**测试结果：**

sg90马达循环转动, 并且串口输入如下：



## IIC SSD1306测试

### 场景描述

利用IIC接口连接SSD1306显示图片。



使用过程中会用到Adafruit\_SSD1306/Adafruit\_GFX库。

* SDA <=> Duo: IIC0\_SDA(PIN2) Duo256: IIC1\_SDA(PIN11)
* SCL <=> Duo: IIC0\_SCL (PIN1) Duo256: IIC1\_SCL(PIN12)
* VCC(红色) <=> 3V3(OUT)
* GND <=> GND

IIC0仅适用于Duo，可以考虑换成其它IIC

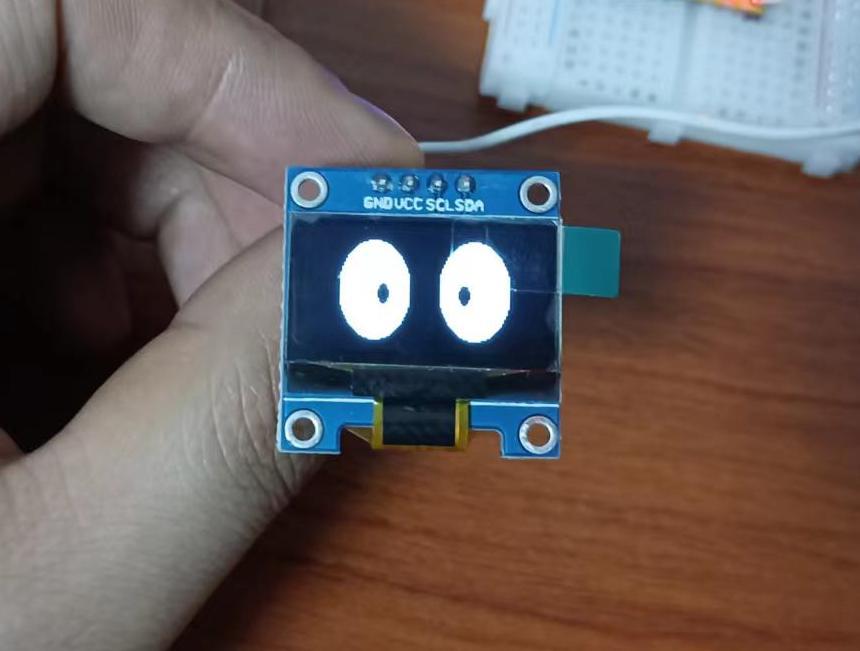
如果要在Duo256上测试，需要更新为最新arduino-sophgo代码

### 测试程序

**见iic\_led.ino**

**测试结果：**

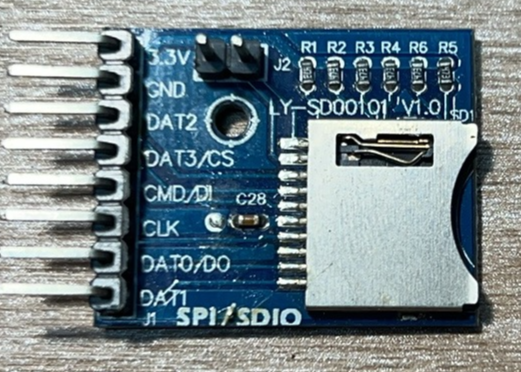
显示动态眼睛画面：



## SPI SD文件系统测试

### 场景描述

使用SPI接口，使用SD转SPI设备，访问SD卡



### 测试程序

**SD read/write测试**

连线：

MOSI - pin 11

MISO - pin 12

CLK - pin 13

CS - pin 4

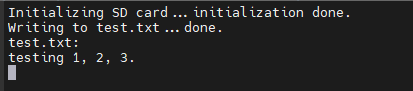
3.3V – 3V3(OUT)

GND - GND

测试需要引入SD库（解压缩library包到arduino-sophgo的library目录下）

程序见sd\_read\_write.ino

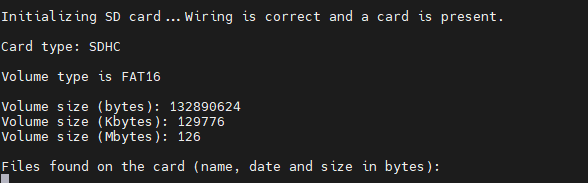
**测试结果：**



**CardInfo测试**

测试见cardinfo.ino

**测试结果：**



## LVGL测试

### 场景描述

LVGL (Light and Versatile Graphics Library) 是最流行的免费开源嵌入式图形库，可为任何 MCU、MPU 等设备创建便捷可用的 UI。本测试使用Duo 的SPI接口和SPI TFT屏幕，运行LVGL的经典测试demo

本测试使用ST7796模块，接线如下：

|  |  |
| --- | --- |
| ST7796 pin | Duo pin |
| CS | 24 |
| DC | 22 |
| RST | 21 |
| BL/LED | 20 |
| MOSI | 10 |
| SCLK | 9 |

由于测试基于TFT\_eSPI库，TFT\_eSPI库中的User\_Setup.h文件需要做相关修改

#define ST7796\_DRIVER

// #define TFT\_MOSI 10 // In some display driver board, it might be written as "SDA" and so on.15

// #define TFT\_SCLK 9 //14

// #define TFT\_MISO -1 // Not connected

#define TFT\_CS 24 // Chip select control pin

#define TFT\_DC 22 // Data Command control pin

#define TFT\_RST 21 // Reset pin (could connect to Arduino RESET pin)

#define TFT\_BL 20 // LED back-light

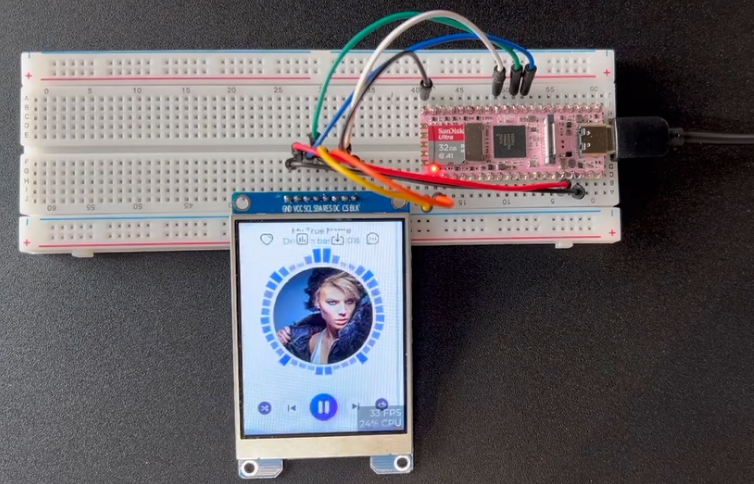
指定相关驱动和PIN连接，如果使用其他GPIO请自行替换，更多配置和添加方法详见

TFT\_eSPI\User\_Setups\SetupX\_Template.h文件

### 测试程序

代码见lvgl.ino （测试需要引入TFT\_eSPI，lvgl等库）

**测试结果：**



#define ST7796\_DRIVER

// #define TFT\_MOSI 10 // In some display driver board, it might be written as "SDA" and so on.15

// #define TFT\_SCLK 9 //14

// #define TFT\_MISO -1 // Not connected

#define TFT\_CS 24 // Chip select control pin

#define TFT\_DC 22 // Data Command control pin

#define TFT\_RST 21 // Reset pin (could connect to Arduino RESET pin)

#define TFT\_BL 20 // LED back-light

指定相关驱动和PIN连接，如果使用其他GPIO请自行替换，更多配置和添加方法详见

TFT\_eSPI\User\_Setups\SetupX\_Template.h文件

### 测试程序

测试程序见lvgl.ino （测试需要引入TFT\_eSPI，lvgl等库）

**测试结果：**

