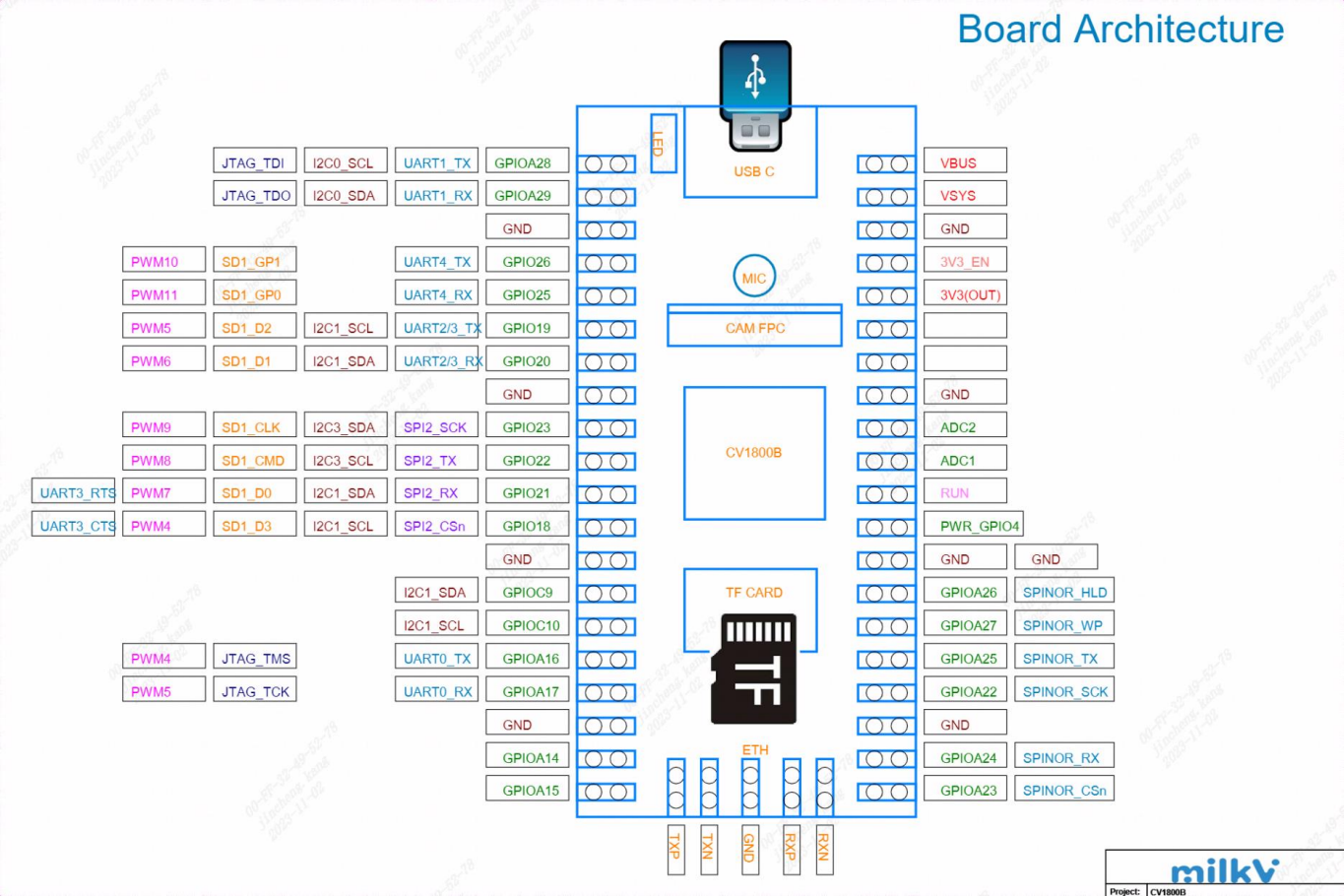


# Milk-V Duo 加载 DF9GMS 舵机

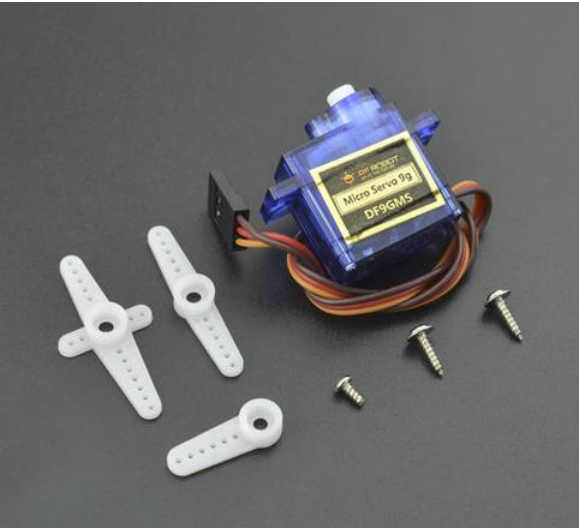
## 一、硬件信息

### duo 开发板引脚

GitHub 链接: <https://github.com/milkv-duo/duo-files/blob/master/duo-schematic-v1.1.pdf>



### DF9GMS 舵机



由 DFRobot 出品的 DF9GMS 微型舵机，该舵机采用高强度 ABS 透明外壳配以内部高精度尼龙齿轮组，加上精准的控制电路、高档轻量化空心杯电机使该微型舵机的重量只有 9 克，而输出力矩达到了惊人的 1.6kg/cm。

技术规格：

工作电压：4.8V

转矩：1.6kg/cm (4.8V)

速度：0.14 秒/60 度 (4.8V)

使用温度：-30~+60 摄氏度

死区宽度：0.5 毫秒

外形尺寸：23x12.2x29mm

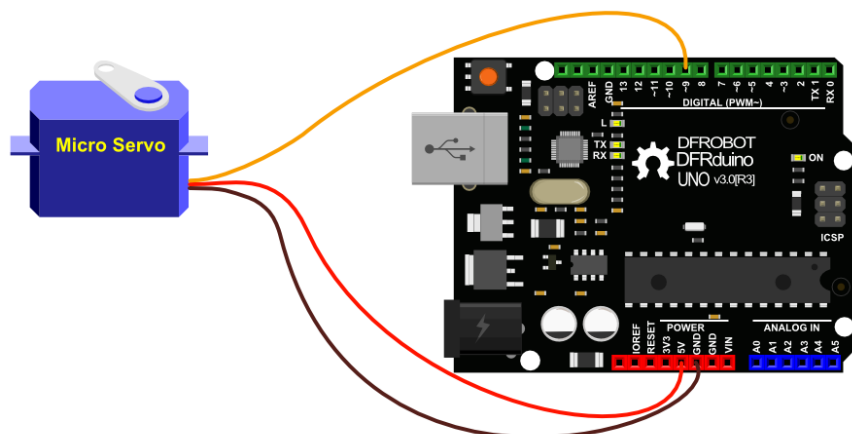
重量：9g

## 组成结构及工作原理



## 连接图

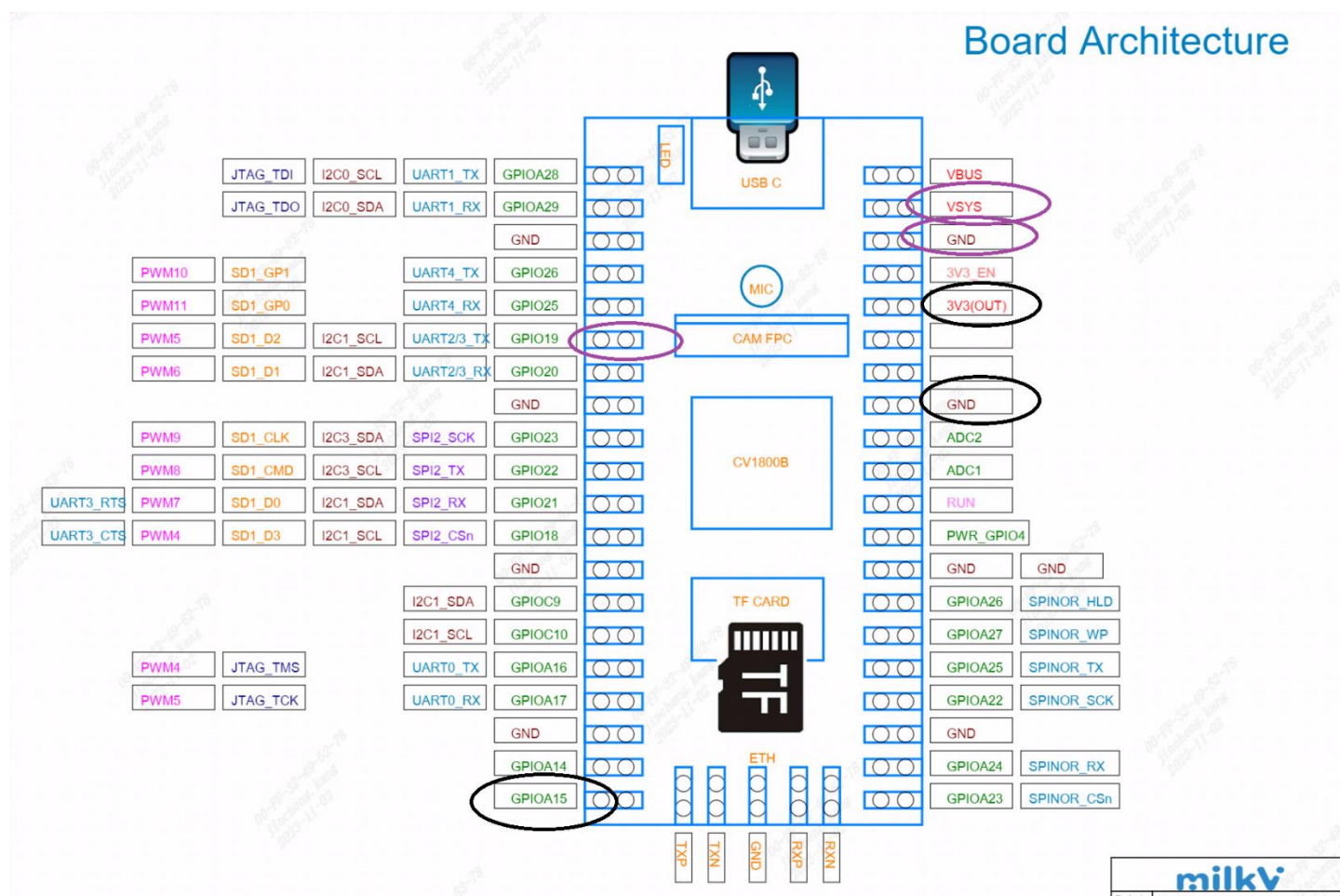
- 硬件
  - 1 x Arduino UNO 控制板
  - 1 x DF9GMS 微型舵机
  - 若干 杜邦线
  - 灰色——GND、红色——VCC、黄色——信号线



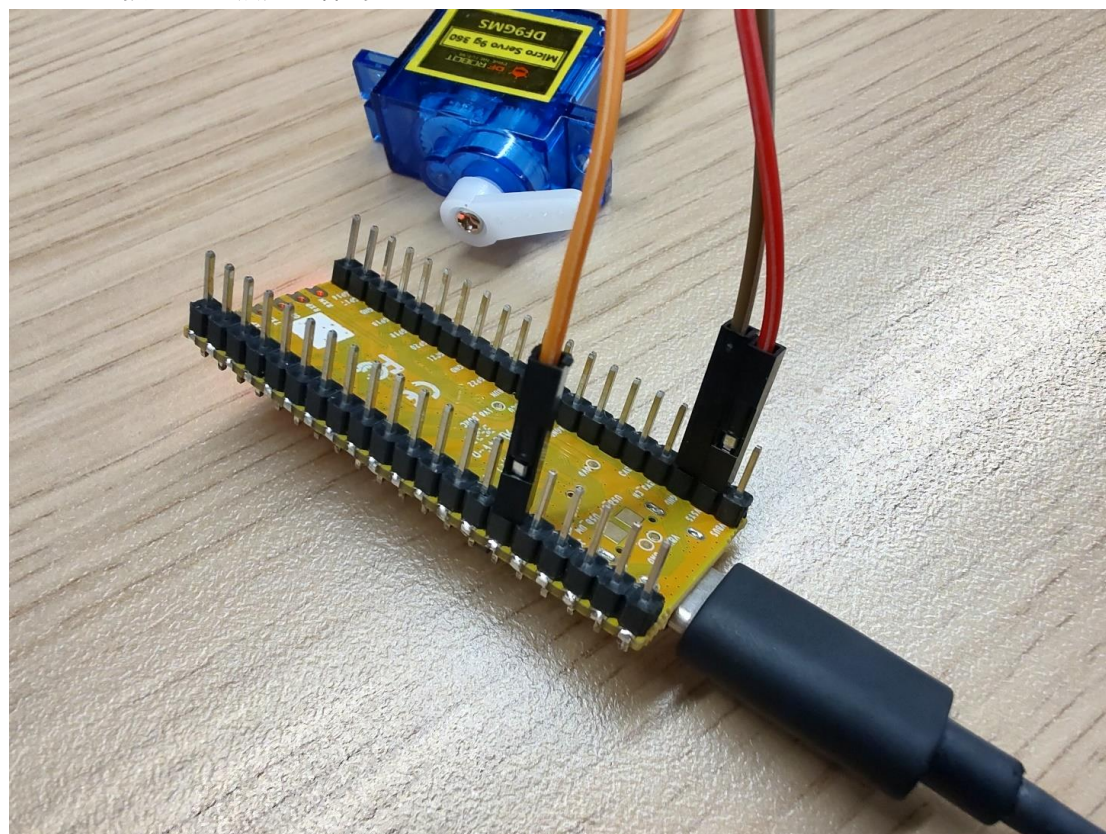
## 连接到开发板上

DF9GMS：红线接 VSYS，棕线接地，橙线接 GPIO19。

电路图如下所示：紫色圆圈表示 DF9GMS。



DF9GMS 接好后应该是这样的：



## 二、示例代码

GitHub 链接: <https://github.com/milkv-duo/duo-examples>

### df9gms.c:

---

```
#include <stdio.h>
#include <unistd.h>
#include <wiringx.h>

/*
    Duo
    -----
    PWM 操作在定频时钟 100MHz, 写 Period 单位为 ns

    DF9GMS 360 度 PWM Duty
    -----
    0.4ms - 1.4ms 顺时针减速
    1.5ms         停止
    1.6ms - 3ms   逆时针加速
*/

static int PWM_PIN = 4; // PWM5@GP4

int main()
{
    long i;

    if(wiringXSetup("duo", NULL) == -1) {
        wiringXGC();
        return -1;
    }

    wiringXSetPWMPeriod(PWM_PIN, 20000000); // 20ms
    wiringXSetPWMDuty(PWM_PIN, 1500000);    // 1.5ms stop
    wiringXSetPWMPolarity(PWM_PIN, 0);      // 0-normal, 1-inversed
    wiringXPWMEnable(PWM_PIN, 1);           // 1-enable, 0-disable

    delayMicroseconds(1000000); // 1s

    for (i = 10000; i < 3000000; i += 10000) // 10 us 步进
    {
        wiringXSetPWMDuty(PWM_PIN, i);
        printf("Duty: %ld\n", i);
        delayMicroseconds(50000); // 50ms
    }

    wiringXSetPWMDuty(PWM_PIN, 1500000);    // 1.5ms stop
```



```
    return 0;
}
```

---

## Makefile:

---

```
TARGET=df9gms
```

```
ifeq (,$(TOOLCHAIN_PREFIX))
$(error TOOLCHAIN_PREFIX is not set)
endif
```

```
ifeq (,$(CFLAGS))
$(error CFLAGS is not set)
endif
```

```
ifeq (,$(LDFLAGS))
$(error LDFLAGS is not set)
endif
```

```
CC = $(TOOLCHAIN_PREFIX)gcc
```

```
CFLAGS += -I$(SYSROOT)/usr/include
```

```
LDFLAGS += -L$(SYSROOT)/lib
LDFLAGS += -L$(SYSROOT)/usr/lib
LDFLAGS += -lwiringx
```

```
SOURCE = $(wildcard *.c)
OBJS = $(patsubst %.c,%.o,$(SOURCE))
```

```
$(TARGET): $(OBJS)
    $(CC) -o $@ $(OBJS) $(LDFLAGS)
```

```
%.o: %.c
    $(CC) $(CFLAGS) -o $@ -c $<
```

```
.PHONY: clean
```

```
clean:
    @rm *.o -rf
    @rm $(OBJS) -rf
    @rm $(TARGET)
```

---

## 三、准备开发环境

1、使用本地的 Ubuntu 系统，推荐 Ubuntu 20.04 LTS(也可以使用虚拟机中的 Ubuntu 系统、Windows 中 WSL 安装的 Ubuntu、基于 Docker 的 Ubuntu 系统)

## 2、安装编译依赖的工具:

```
sudo apt-get install wget git make
```

## 3、获取 Examples

```
git clone https://github.com/milkv-duo/duo-examples.git
```

## 4、加载编译环境

```
cd duo-examples
source envsetup.sh
```

5、第一次加载会自动下载所需的 SDK 包，大小为 180M 左右，下载完会自动解压到 duo-examples 下，解压后的目录名为 duo-sdk，下次加载时检测到已存在该目录，就不会再次下载了。

注: 如果因为网络原因无法完成 SDK 包的下载，请通过其他途径获取到 duo-sdk.tar.gz 包，手动解压到 duo-examples 目录下，重新 source envsetup.sh。下载路径在 envsetup.sh 中能找到: <https://github.com/milkv-duo/duo-app-sdk/releases/download/duo-app-sdk-v1.2.0/duo-sdk-v1.2.0.tar.gz> 如下图所示。

```
1  #!/bin/bash
2
3  SDK_DIR=$(cd -- "$(dirname -- "${BASH_SOURCE[0]}")" &> /dev/null && pwd)
4  echo "SDK_DIR: ${SDK_DIR}"
5
6  MILKV_DUO_SDK=${SDK_DIR}/duo-sdk
7  TOOLCHAIN_DIR=${MILKV_DUO_SDK}/riscv64-linux-musl-x86_64
8
9  SDK_URL="https://github.com/milkv-duo/duo-app-sdk/releases/download/duo-app-sdk-v1.2.0/duo-sdk-v1.2.0.tar.gz"
```

## 6、编译测试

以 hello-world 为例，进入该例子目录直接执行 make 即可

```
cd hello-world
make
```

编译成功后将生成的 helloworld 可执行程序通过网口或者 RNDIS 网络等方式传送到 Duo 设备中，比如[默认固件](#)支持的 RNDIS 方式，Duo 的 IP 为 192.168.42.1，用户名是 root，密码是 milkv

```
$ scp helloworld root@192.168.42.1:/root/
```

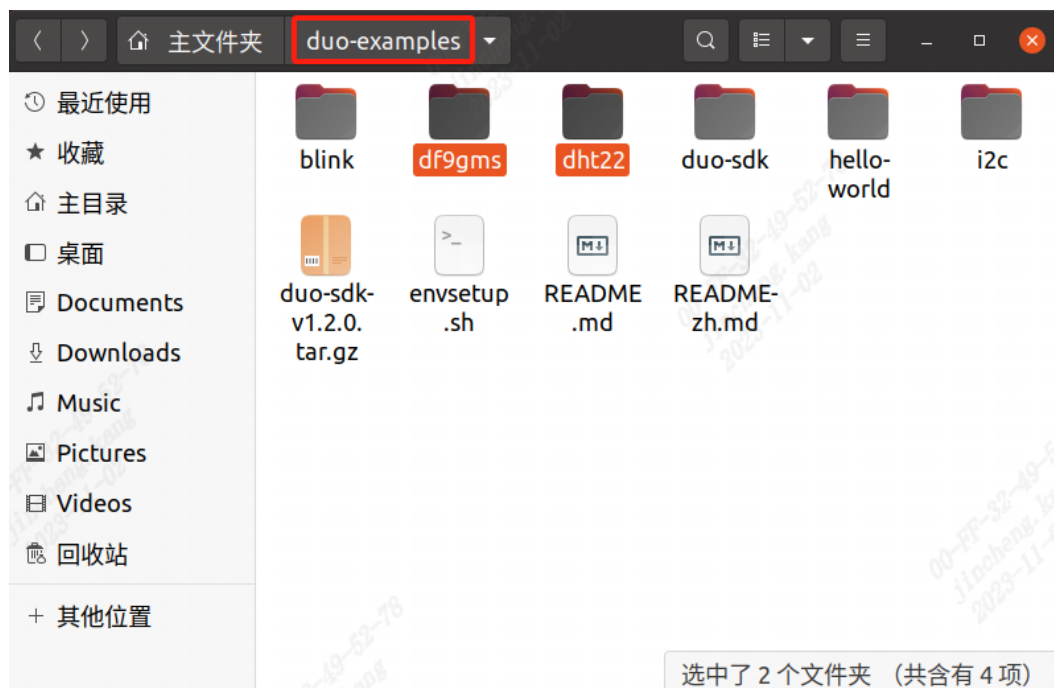
发送成功后，在 ssh 或者串口登陆的终端中运行 ./helloworld，会打印 Hello, World!

```
[root@milkv]~# ./helloworld
Hello, World!
```

7、至此，我们的编译开发环境就可以正常使用了。

# 四、运行程序

如图所示:



接下来开始编译，以 df9gms 为例，进入该例子目录直接执行 make 即可

```
cd df9gms
make
```

make 报错 source 一下即可。编译成功后得到 df9gms 可执行程序。如下图所示。

```
kjc@ubuntu: ~/duo-examples/df9gms
kjc@ubuntu:~/duo-examples$ source envsetup.sh
SDK_DIR: /home/kjc/duo-examples
SDK environment is ready
kjc@ubuntu:~/duo-examples$ cd df9gms/
kjc@ubuntu:~/duo-examples/df9gms$ ls
df9gms.c  Makefile
kjc@ubuntu:~/duo-examples/df9gms$ make
/home/kjc/duo-examples/duo-sdk/riscv64-linux-musl-x86_64/bin/riscv64-unknown-linux-musl-gcc -D_LARGEFILE_SOURCE -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -I/home/kjc/duo-examples/duo-sdk/rootfs/usr/include -o df9gms.o -c df9gms.c
/home/kjc/duo-examples/duo-sdk/riscv64-linux-musl-x86_64/bin/riscv64-unknown-linux-musl-gcc -o df9gms df9gms.o -mcpu=c906fdv -march=rv64imafdcv0p7xthead -mcmode l=medany -mabi=lp64d -L/home/kjc/duo-examples/duo-sdk/rootfs/lib -L/home/kjc/duo-examples/duo-sdk/rootfs/usr/lib -lwiringx
kjc@ubuntu:~/duo-examples/df9gms$ ls
df9gms df9gms.c df9gms.o Makefile
kjc@ubuntu:~/duo-examples/df9gms$
```

然后把 df9gms 上传到开发板 root 路径下，输入 ./df9gms 即可运行。运行成功截图如下：

```
[root@milkv-duo]~# ./df9gms
```

```
Duty: 10000
```

```
Duty: 20000
```

```
Duty: 30000
```

```
Duty: 40000
```

```
Duty: 50000
```

```
Duty: 60000
```

```
Duty: 70000
```

```
Duty: 80000
```

```
Duty: 90000
```

```
Duty: 100000
```

```
Duty: 110000
```

```
Duty: 120000
```

```
Duty: 130000
```

```
Duty: 140000
```

```
Duty: 150000
```