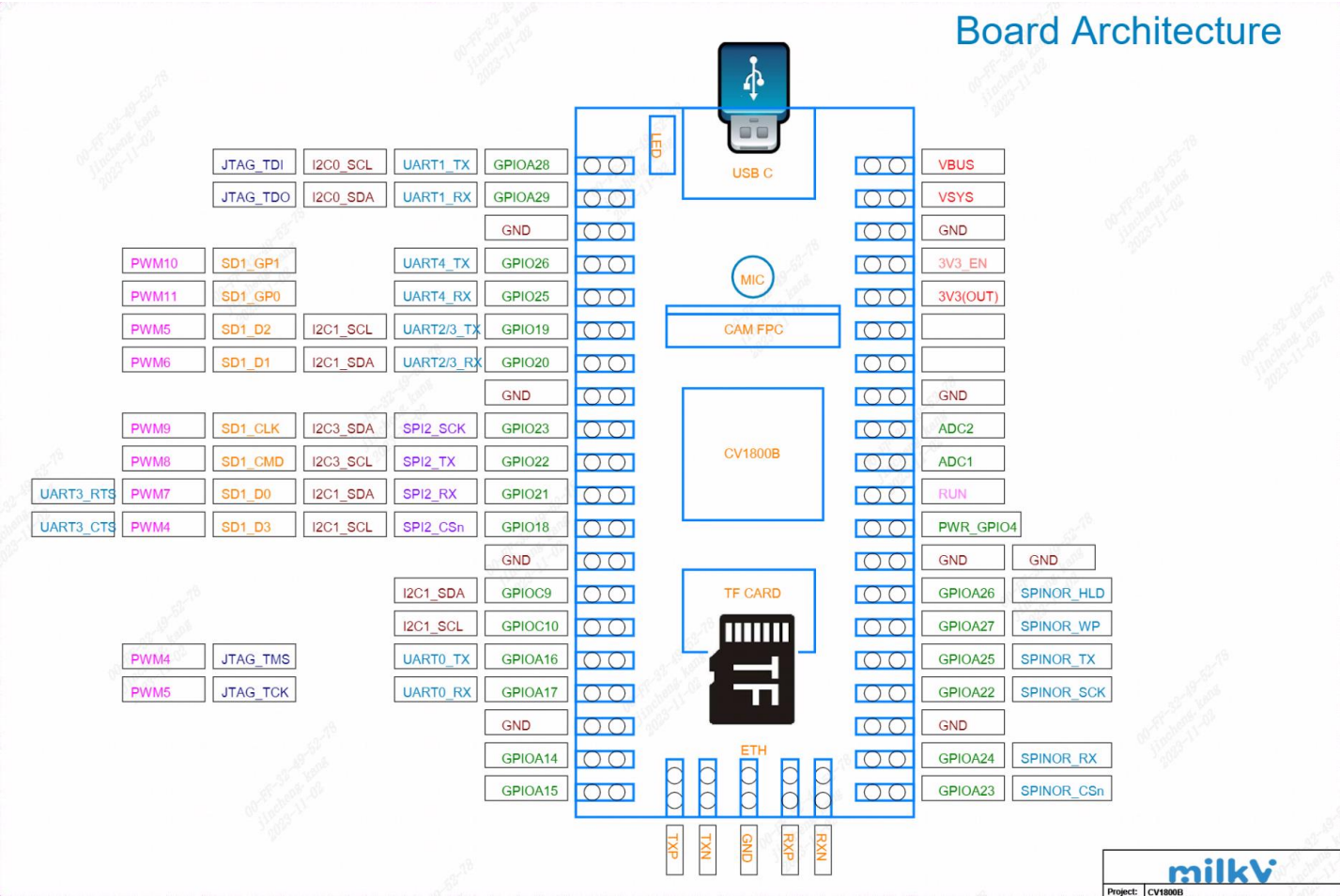


Milk-V Duo 加载 DHT22 温湿度传感器

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>



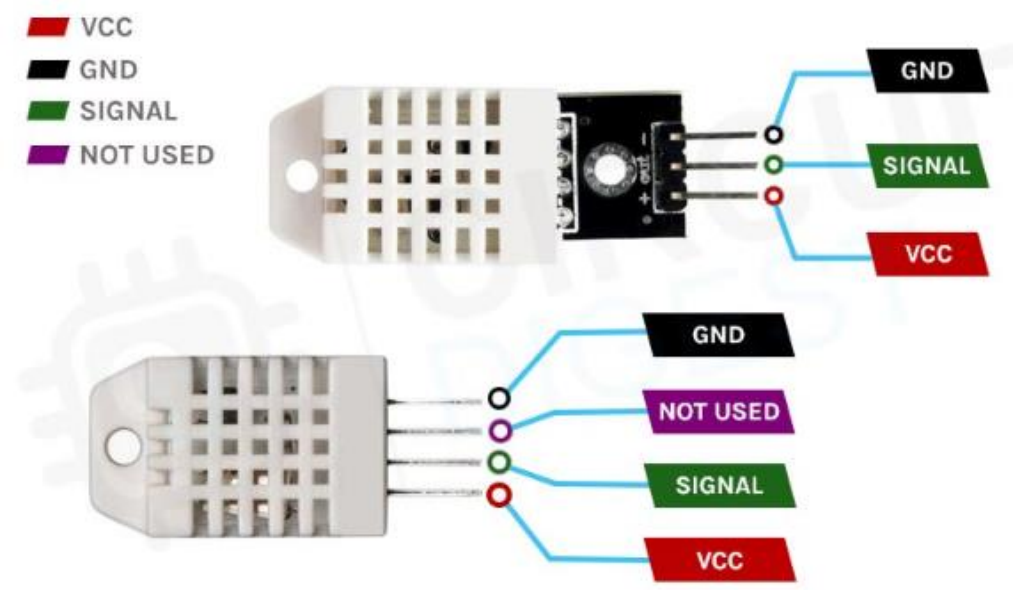
DHT22 温湿度传感器

dfrobotwiki 链接如下：
https://wiki.dfrobot.com.cn/SKU_SEN0137_%E9%AB%98%E7%B2%BE%E5%BA%A6DHT22%E6%B8%A9%E6%B9%BF%E5%BA%A6%E4%BC%A0%E6%84%9F%E5%99%A8

DHT22 温湿度传感器是一款多功能且经济高效的传感器，用于测量环境温度和湿度，应用范围广泛。它基于数字信号输出，可以提供高精度测量，温度分辨率为 0.1 摄氏度，湿度分辨率为 0.1%。该传感器采用电容式湿度传感元件和热敏电阻分别测量湿度和温度。DHT22 传感器的功耗也相对较低，可在 3.3V 至 5V 的电压范围内工作，使其适用于电池供电项目。此外，该传感器具有长期稳定性和高可靠性，这使其成为暖通空调等各种应用的完美选择，气象站和室内空气质量监测系统。

传感器模块

DHT22 模块共有 3 个引脚。而 DHT22 裸骨传感器有 4 个引脚。如果我们考虑三个引脚中的模块，其中两个是电源引脚，一个是数据引脚。如果我们查看 4 针传感器，则额外的针是 NC 针，它没有任何功能。模块和传感器的引脚图如下所示。



- DATA 用于 1-Wire 通信的数据引脚。
- GND 模块的接地引脚。
- VCC 模块的电源引脚。
- Not Used 在此传感器中，未使用此引脚。

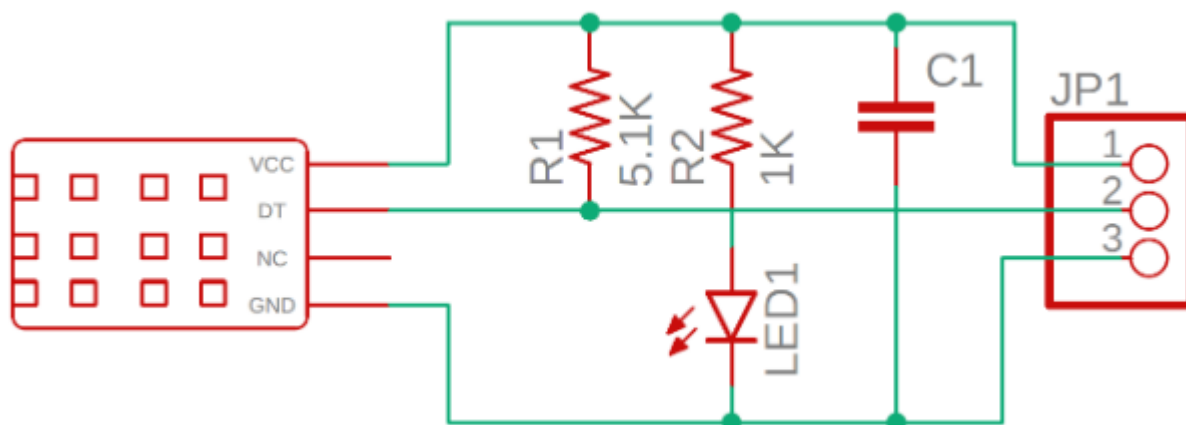
DHT22 传感器模块的零件标记

除传感器外，DHT22 模块在 PCB 上仅包含两个组件。一个上拉电阻和一个去耦电容，DHT22 模块的零件标记如下所示。



DHT22 模块电路图

DHT22 温湿度传感器模块完整原理图如下图：



DHT22 模块的原理图如上所示。如前所述，董事会只有几名成员。VCC 和 GND 引脚直接连接到 DHT22，上拉电阻连接到 DATA 引脚。钽电容和多层电容提供足够的滤波。作为电源指示，在某些 PCB 中，您可以找到 LED 指示灯，但对于大多数电路板来说，LED 是不存在的。

DHT22 传感器模块常见问题

问：简而言之，DHT22 是什么？

DHT22 是 DHT11 传感器的更昂贵版本，显然具有更好的规格。其温度测量范围为 -40 至 +125 摄氏度，精度为 ± 0.5 度，而 DHT11 温度范围为 0 至 50 摄氏度，精度为 ± 2 度。

问：DHT22 是模拟的还是数字的？

DHT-22（也称为 AM2302）是一种数字输出相对湿度和温度传感器。

问：DHT22 防水吗？

不，它不防水。

问：DHT11 传感器的采样率是多少？

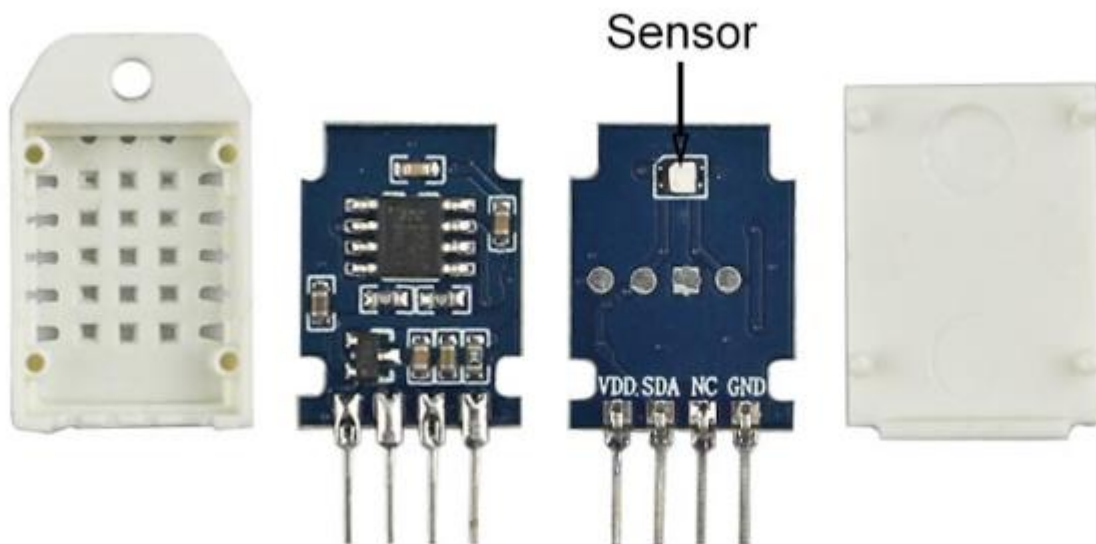
DHT22 的采样率为 1Hz。

问：DHT22 使用什么协议？

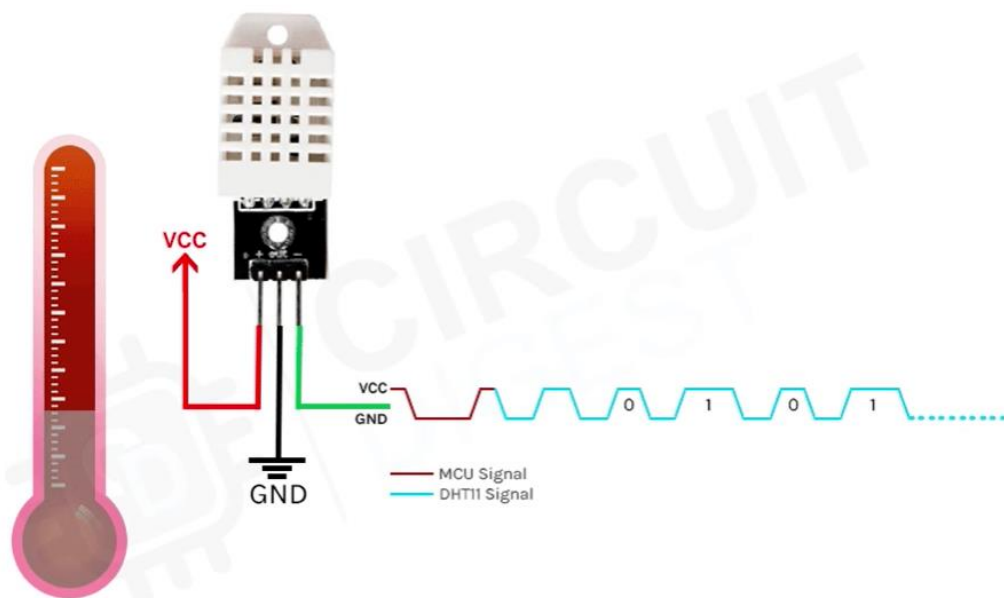
DHT22 传感器使用专用的单总线通信协议，可以通过计算的定时脉冲发送和接收数据。

DHT22 是如何工作的？

如果您使用的是原装 DHT22 传感器，那么它里面会有一个 NTC 热敏电阻和传感器模块，但是您在市场上可以找到的大多数传感器大多是非原装零件，并且在里面您会发现一个小传感器，您可以在下图中看到。



湿度传感元件由夹在两个电极之间的保湿基板组成。当基材吸收水分时，两个电极之间的电阻会降低。两个电极之间的电阻变化与相对湿度成正比。较高的相对湿度会降低电极之间的电阻，而较低的相对湿度会增加电极之间的电阻。这种电阻变化是通过板载 MCU 的 ADC 测量的，并计算出相对湿度。

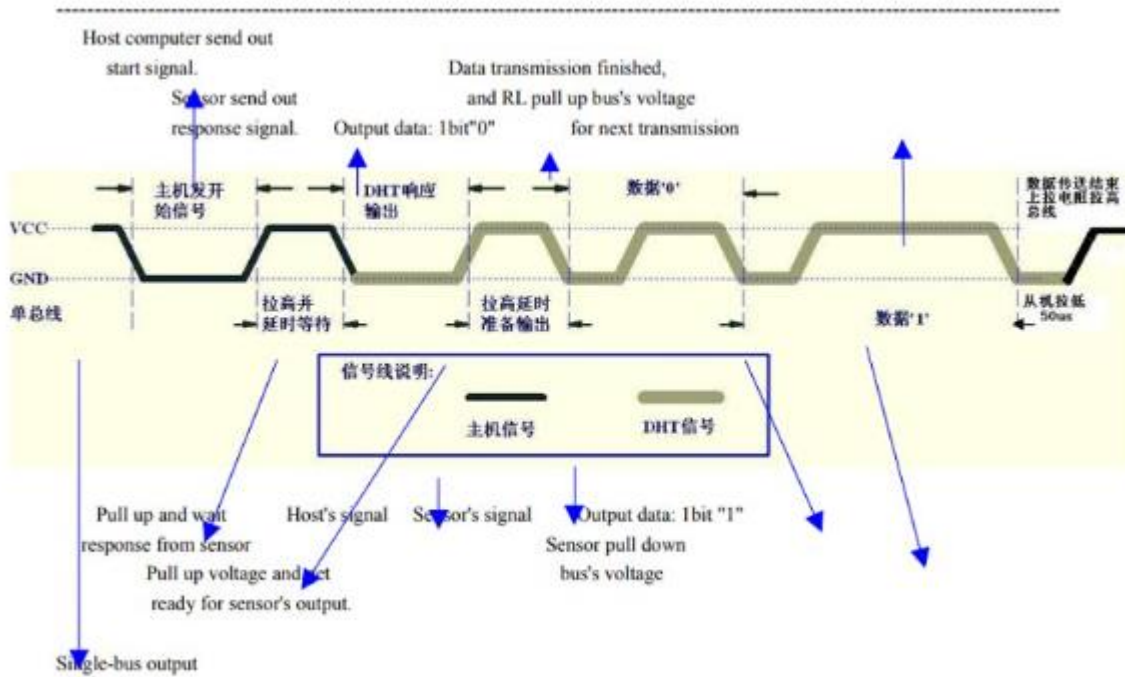


每个 DHT22 元件都经过实验室严格校准，湿度校准极其准确。校准系数作为程序存储在 OTP 存储器中，供传感器内部信号检测过程使用。

DHT22 单总线通信协议

单总线通信协议用于与 DHT22 和微控制器通信。样本数据大约需要 4 毫秒才能运行完毕。此数据由小数部分和整数部分组成。总数据为 40 位长，为 MSB 格式。数据格式如下：8bit 整数 RH 数据+8bit 十进制 RH 数据+8bit 整数 T 数据+8bit 十进制 T 数据+8bit 校验和。如果数据传输正确，则校验和应为“8bit 整数 RH 数据+8bit 十进制 RH 数据+8bit 整数 T 数据+8bit 十进制 T 数据”的最后 8bit。

当 MCU 发送启动信号 DHT 从低功耗模式变为运行模式，并将所有 40 位数据转储给微控制器，微控制器读取数据并根据二进制数据计算温度和湿度。

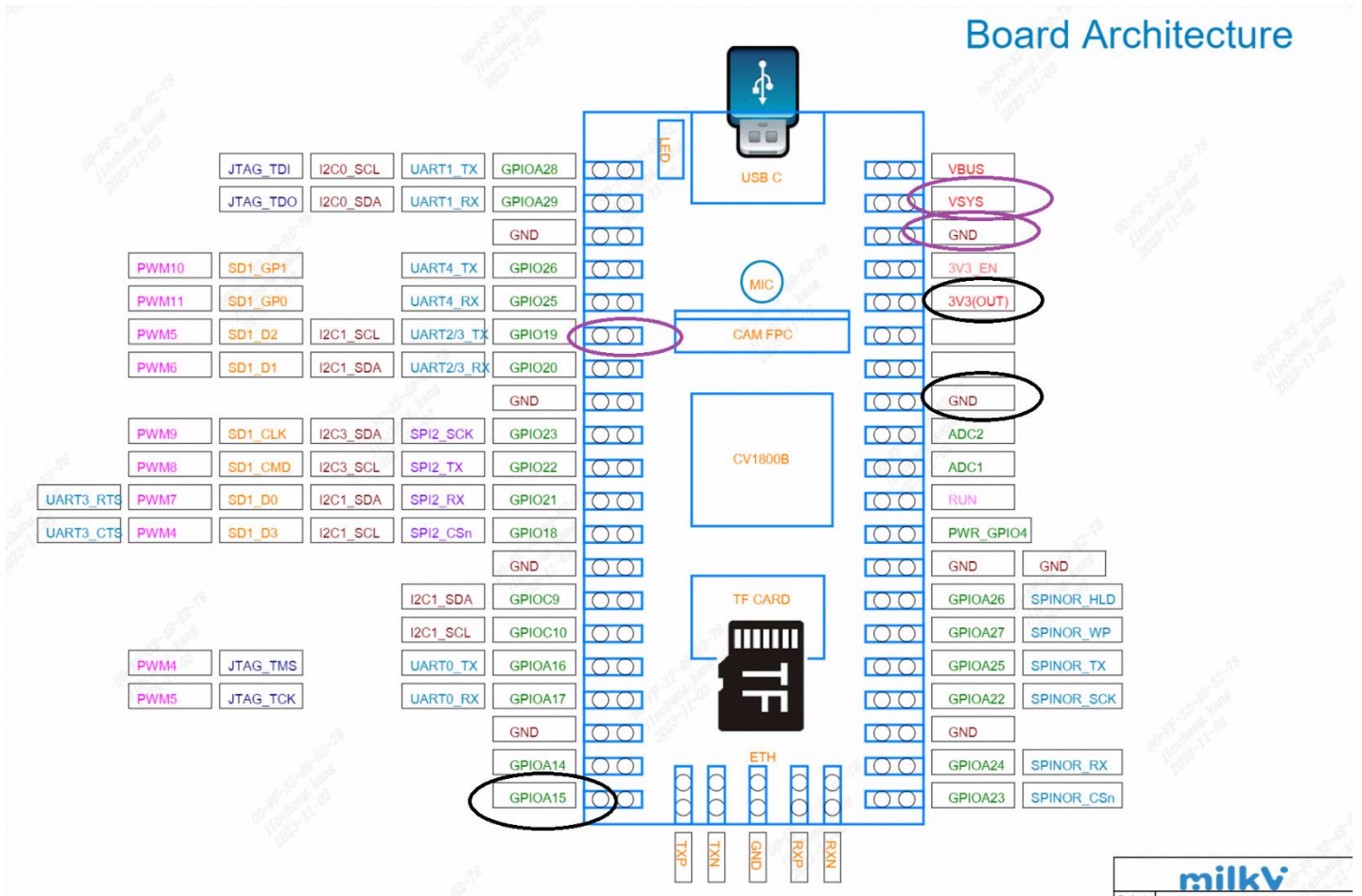


上图显示了数据通信如何与微控制器和 DHT22 一起工作。

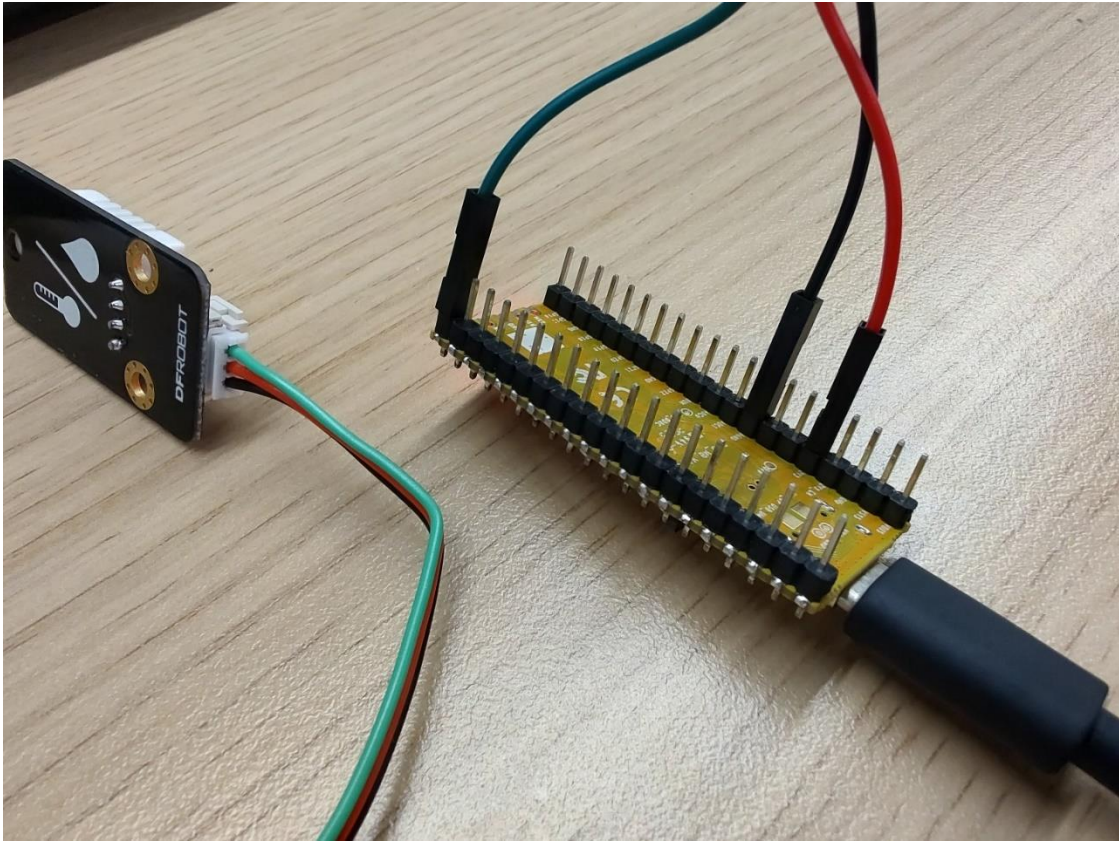
连接到开发板上

DHT22: 红线接 3V3(OUT), 黑线接地, 绿线接 GPIOA15。

电路图如下所示: 黑色圆圈表示 DHT22。



DHT22 接好后应该是这样的:



二、示例代码

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

dht22.c:

```
// Ref: https://github.com/technion/loI_dht22/blob/master/dht22.c
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <unistd.h>
#include <wiringx.h>
#define MAXTIMINGS 85
static int DHTPIN = 15;
static int dht22_dat[5] = {0, 0, 0, 0, 0};
static uint8_t sizecvt(const int read)
{
    /* digitalRead() and friends from wiringpi are defined as returning a value
    < 256. However, they are returned as int() types. This is a safety function */

    if (read > 255 || read < 0)
    {
        printf("Invalid data from wiringPi library\n");
        exit(EXIT_FAILURE);
    }
    return (uint8_t)read;
}
```

```

static int read_dht22_dat()
{
    uint8_t laststate = HIGH;
    uint8_t counter = 0;
    uint8_t j = 0, i;

    dht22_dat[0] = dht22_dat[1] = dht22_dat[2] = dht22_dat[3] = dht22_dat[4] = 0;

    // pull pin down for 18 milliseconds
    pinMode(DHTPIN, PINMODE_OUTPUT);
    digitalWrite(DHTPIN, HIGH);
    // delay(500);
    delayMicroseconds(500000);
    digitalWrite(DHTPIN, LOW);
    // delay(20);
    delayMicroseconds(20000);
    // prepare to read the pin
    pinMode(DHTPIN, PINMODE_INPUT);

    // detect change and read data
    for (i = 0; i < MAXTIMINGS; i++)
    {
        counter = 0;
        while (sizecvt(digitalRead(DHTPIN)) == laststate)
        {
            counter++;
            delayMicroseconds(2);
            if (counter == 255)
            {
                break;
            }
        }
        laststate = sizecvt(digitalRead(DHTPIN));

        if (counter == 255)
            break;

        // ignore first 3 transitions
        if ((i >= 4) && (i % 2 == 0))
        {
            // shove each bit into the storage bytes
            dht22_dat[j / 8] <= 1;
            if (counter > 16)
                dht22_dat[j / 8] |= 1;
            j++;
        }
    }

    // check we read 40 bits (8bit x 5 ) + verify checksum in the last byte

```

```

// print it out if data is good
if ((j >= 40) &&
    (dht22_dat[4] == ((dht22_dat[0] + dht22_dat[1] + dht22_dat[2] + dht22_dat[3]) & 0xFF)))
{
    float t, h;
    h = (float)dht22_dat[0] * 256 + (float)dht22_dat[1];
    h /= 10;
    t = (float)(dht22_dat[2] & 0x7F) * 256 + (float)dht22_dat[3];
    t /= 10.0;
    if ((dht22_dat[2] & 0x80) != 0)
        t *= -1;
    printf("Humidity = %.2f %% Temperature = %.2f *C \n", h, t);
    return 1;
}
else
{
    printf("Data not good, skip\n");
    return 0;
}
}

int main()
{
    if (wiringXSetup("duo", NULL) == -1)
    {
        wiringXGCC();
        return -1;
    }

    if (wiringXValidGPIO(DHTPIN) != 0)
    {
        printf("Invalid GPIO %d\n", DHTPIN);
    }

    while (1)
    {
        read_dht22_dat();
        delayMicroseconds(1500000);
    }

    return 0;
}

```

Makefile:

```
TARGET=dht22
```

```

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

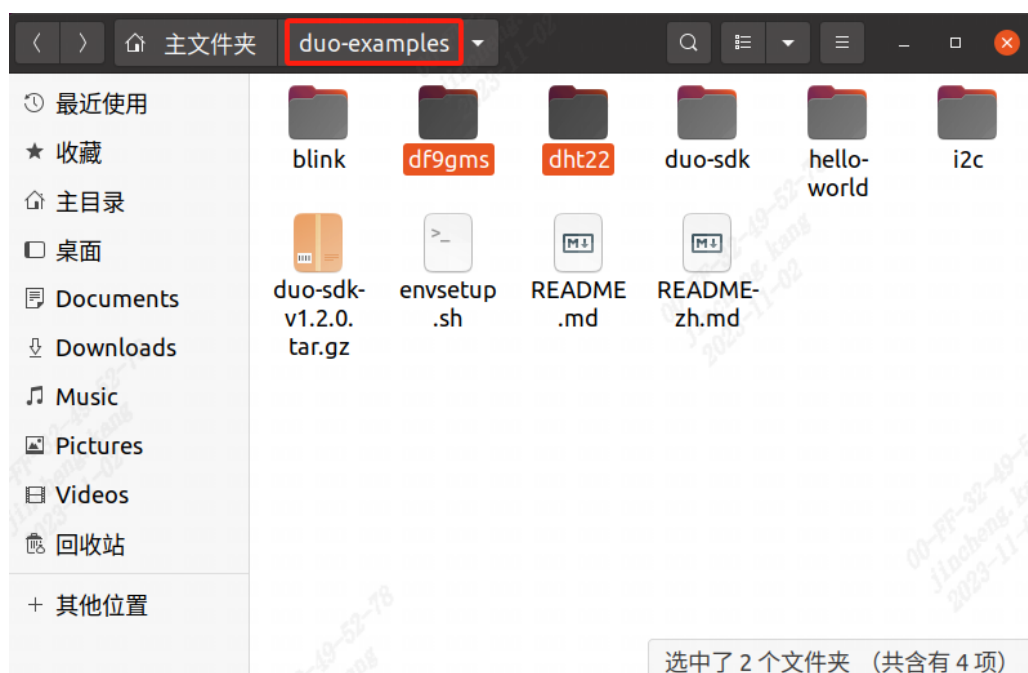
```
$ scp helloworld root@192.168.42.1:/root/
```

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。

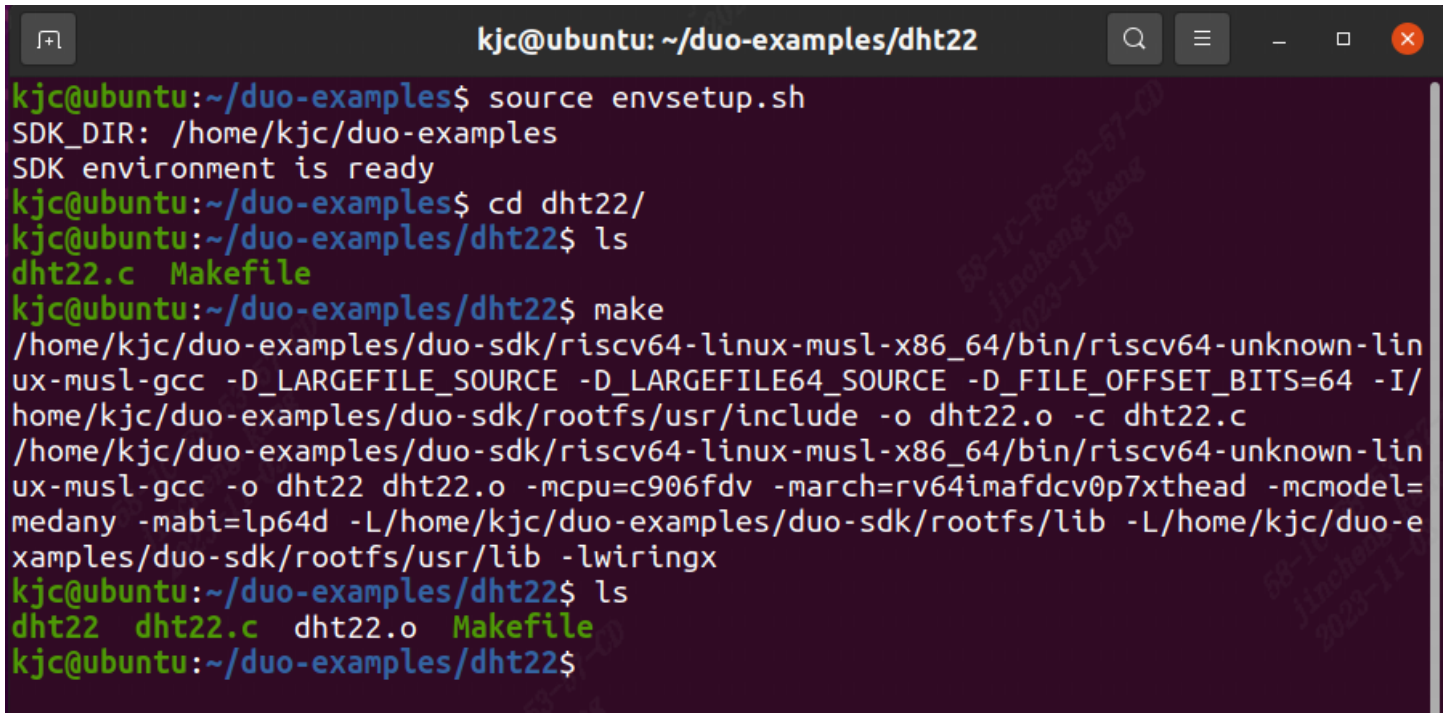
四、运行程序



接下来开始编译，进入该目录直接执行 make 即可

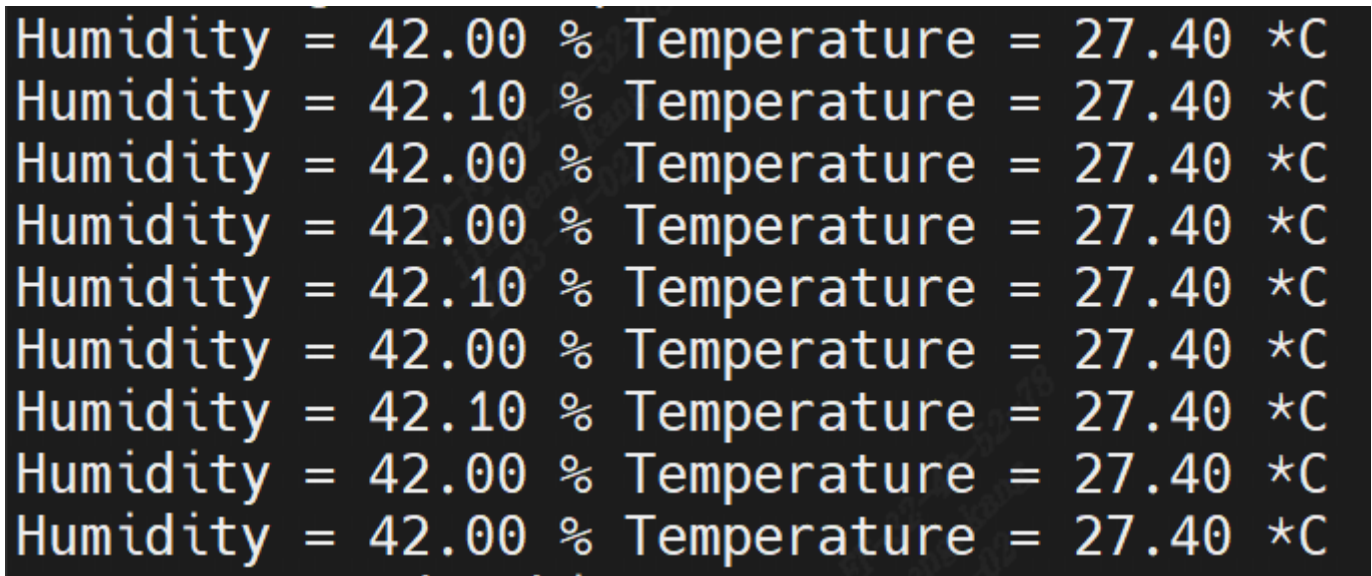
```
cd dht22  
make
```

make 报错 source 一下即可。编译成功后得到 dht22 可执行程序。如下图所示。



```
kjc@ubuntu: ~/duo-examples/dht22  
kjc@ubuntu:~/duo-examples$ source envsetup.sh  
SDK_DIR: /home/kjc/duo-examples  
SDK environment is ready  
kjc@ubuntu:~/duo-examples$ cd dht22/  
kjc@ubuntu:~/duo-examples/dht22$ ls  
dht22.c  Makefile  
kjc@ubuntu:~/duo-examples/dht22$ make  
/home/kjc/duo-examples/duo-sdk/riscv64-linux-musl-x86_64/bin/riscv64-unknown-lin  
ux-musl-gcc -D_LARGEFILE_SOURCE -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -I/  
home/kjc/duo-examples/duo-sdk/rootfs/usr/include -o dht22.o -c dht22.c  
/home/kjc/duo-examples/duo-sdk/riscv64-linux-musl-x86_64/bin/riscv64-unknown-lin  
ux-musl-gcc -o dht22 dht22.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/dht22$ ls  
dht22  dht22.c  dht22.o  Makefile  
kjc@ubuntu:~/duo-examples/dht22$
```

然后把 dht22 上传到开发板 root 路径下，输入 ./dht22 即可运行。运行成功截图如下：



```
Humidity = 42.00 % Temperature = 27.40 *C  
Humidity = 42.10 % Temperature = 27.40 *C  
Humidity = 42.00 % Temperature = 27.40 *C  
Humidity = 42.00 % Temperature = 27.40 *C  
Humidity = 42.10 % Temperature = 27.40 *C  
Humidity = 42.00 % Temperature = 27.40 *C  
Humidity = 42.10 % Temperature = 27.40 *C  
Humidity = 42.00 % Temperature = 27.40 *C  
Humidity = 42.00 % Temperature = 27.40 *C
```