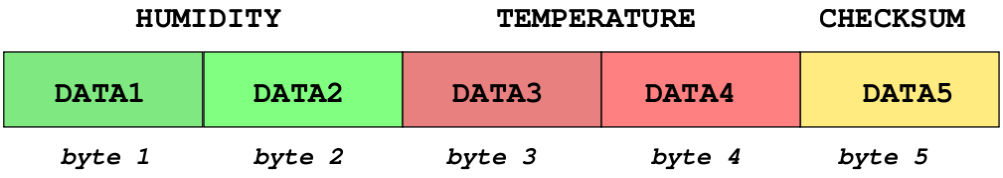
Assignment 2: DHT11 Sensor Notes

Tech Specs

Power: 3.3 V ~ 5.5 V DC
Output: Serial
Measuring Range: (relative) humidity 0 ~ 100%, temperature -40 ~ 120 ℃
There are three sensor pins for Power, Ground and Data transmission, respectively. These pins should be connected to the RPi GPIO's as appropriate. Note there is only one line for data communication between the DHT11 and the RPi. This communication is performed bit-serially in an asynchronous manner, no clocks involved.

Data Sequence

The sensor captures both temperature and humidity from the environment. The DHT11 is equipped with an embedded microcontroller which converts the raw sensing signals into the 5-byte digital format shown below.



Sensor Transaction Data Format

1. 8-bit RH (relative humidity) integer part (DATA1)
2. 8-bit RH decimal part (DATA2)
3. 8-bit temperature integer part (DATA3)
4. 8-bit temperature decimal part (DATA4)
5. 8-bit checksum

This data format encapsulates one temperature/humidity reading of the sensor.
If the data transmission is correct, the checksum should equal the lower 8 bits of the binary addition result:

DATA1 + DATA2 + DATA3 + DATA4

Note that the decimal bytes of DHT11 are always 0. They are intended for future updates.

Communication Transaction Process

The RPi initiate a communication session with DHT11 using a sort of Request/Acknowledge protocol. Basicaly, Ppi request a transaction with the sensor to read its data. The sensor acknowledges and bgins serially transmitting data bits. However, the data bits ‘0’ and ‘1’ are encoded in variable time length pulses of specific duration. Moreover, the bits are separated by 80ns low voltage pulses. RPi will need to interpret these pulses to identify the bit values.

1. **RPi sends start request to sensor**

When the sensor data bus (single wire) is at high voltage, it is in free status. When issuing communication, RPi needs to pull down the voltage of the bus, and hold it at low level at least 500 µs. Then, the data voltage should be pull up again by RPi. To generate a Request (start) signal, RPi needs to hold the bus at high for about 20-40 µs in order for the sensor to respond. (See fFgure 1.)

2. **Sensor responds (acknowledges) request signal from RPi**

When the sensor detects a start signal, it will pull the bus down low and send out a low voltage signal which will last 80 µs. Then sensor will pull up the bus to high voltage for 80 µs, which means a message (Acknowledge) to RPi that the sensor is going to send data. (See Figure 2.)

3. **Sensor begins sending data**

When sensor keeps on sending data to RPi bit-serially. The data bits are separated by 50 µs low voltage. The length of high voltage represents bit values interpreted as follows: (See Figure 3.)

Bit	‘0’	26 - 28 µs
Bit	‘1’	70 µs

The transaction is terminated when all 5 bytes of data is read by RPi. The sensor does not seem to provide any specific termination signal.

The overall transaction communication is shown in Figure 4.

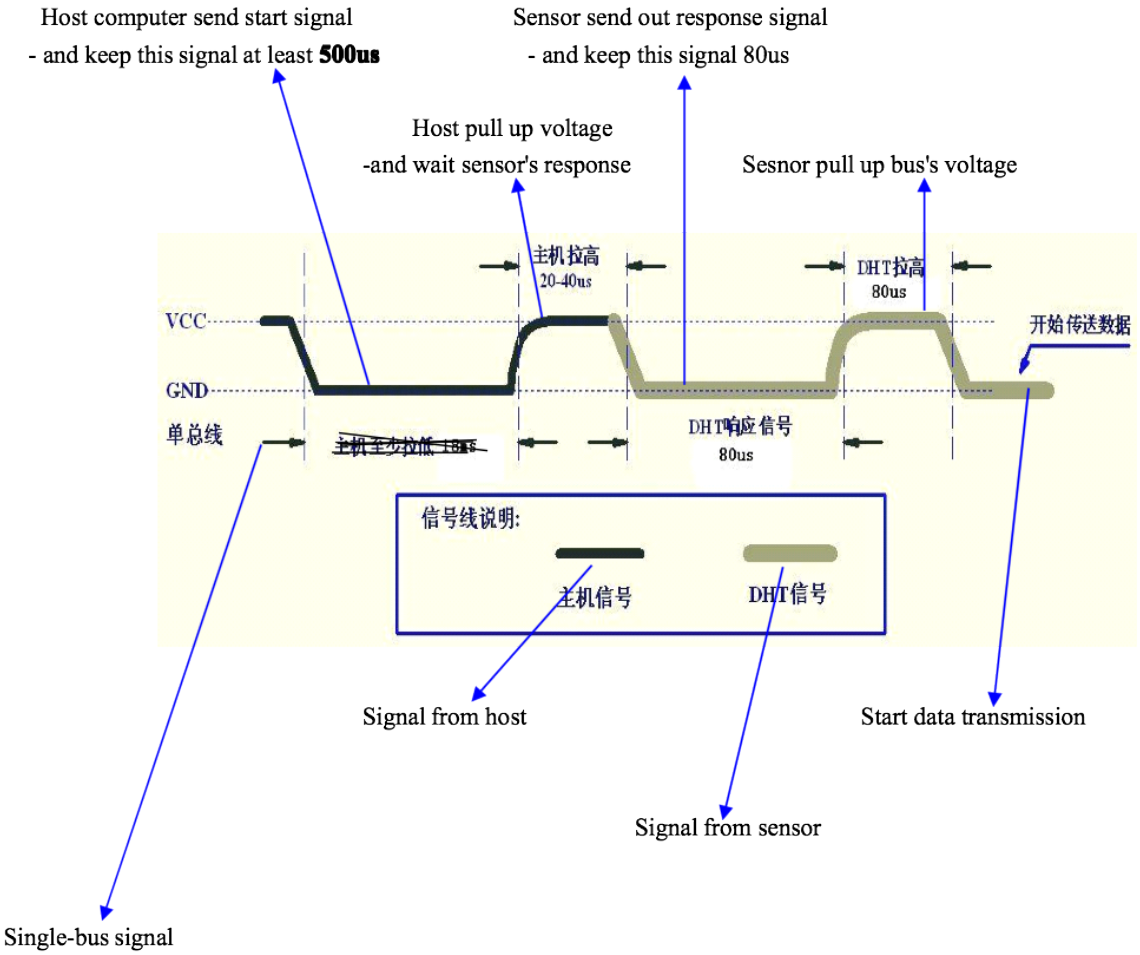


Figure 1: Initiate request/acknowledge signals for transaction

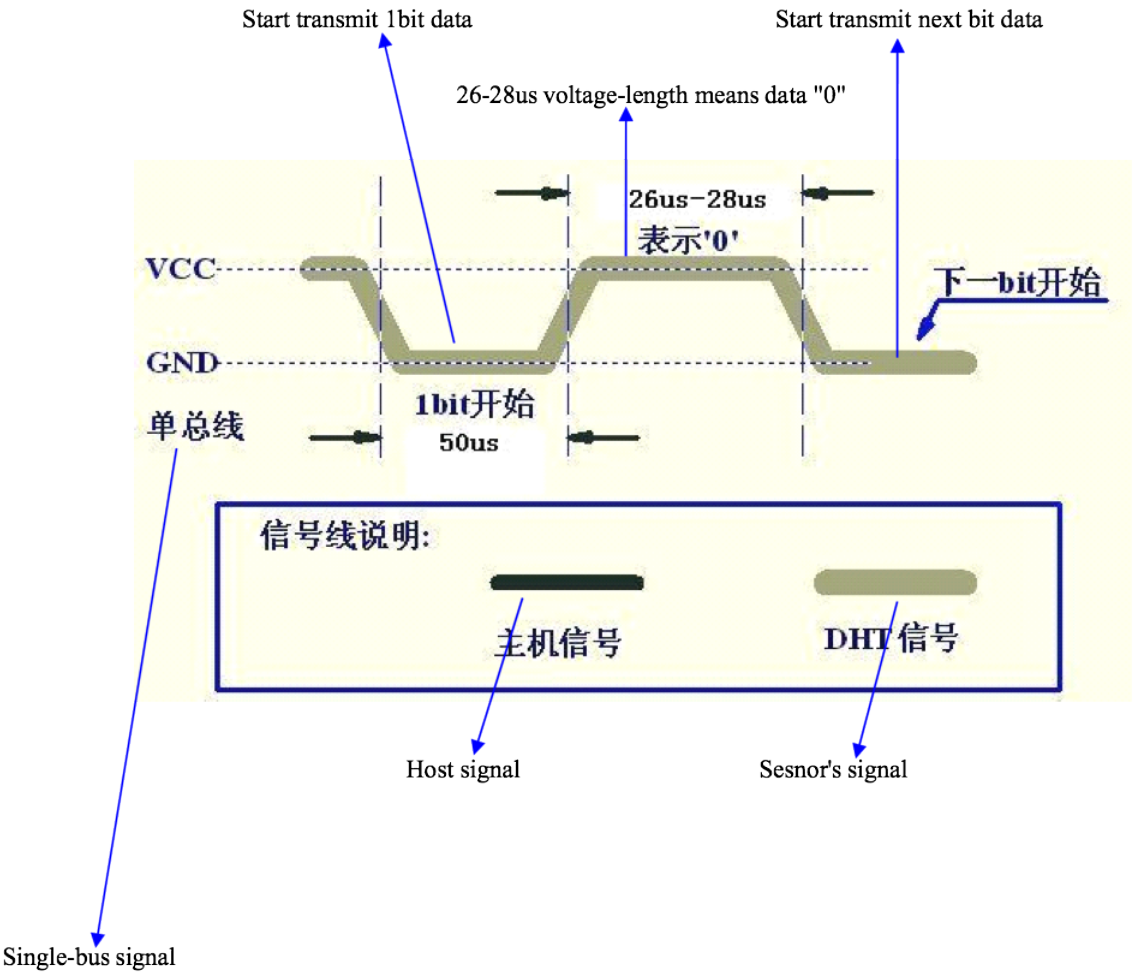


Figure 2: Sensor sending bit ‘0’ (50ns separation)

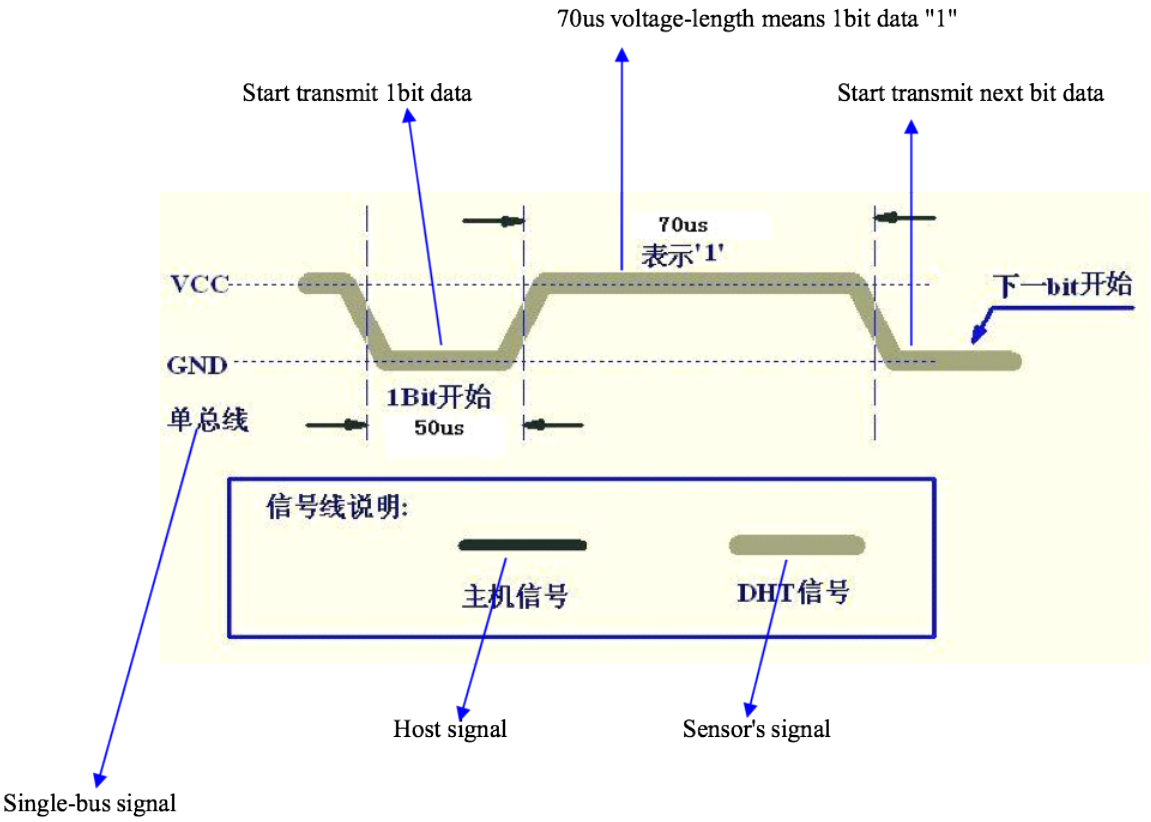


Figure 3: Sensor sending bit ‘1’ (50ns separation)

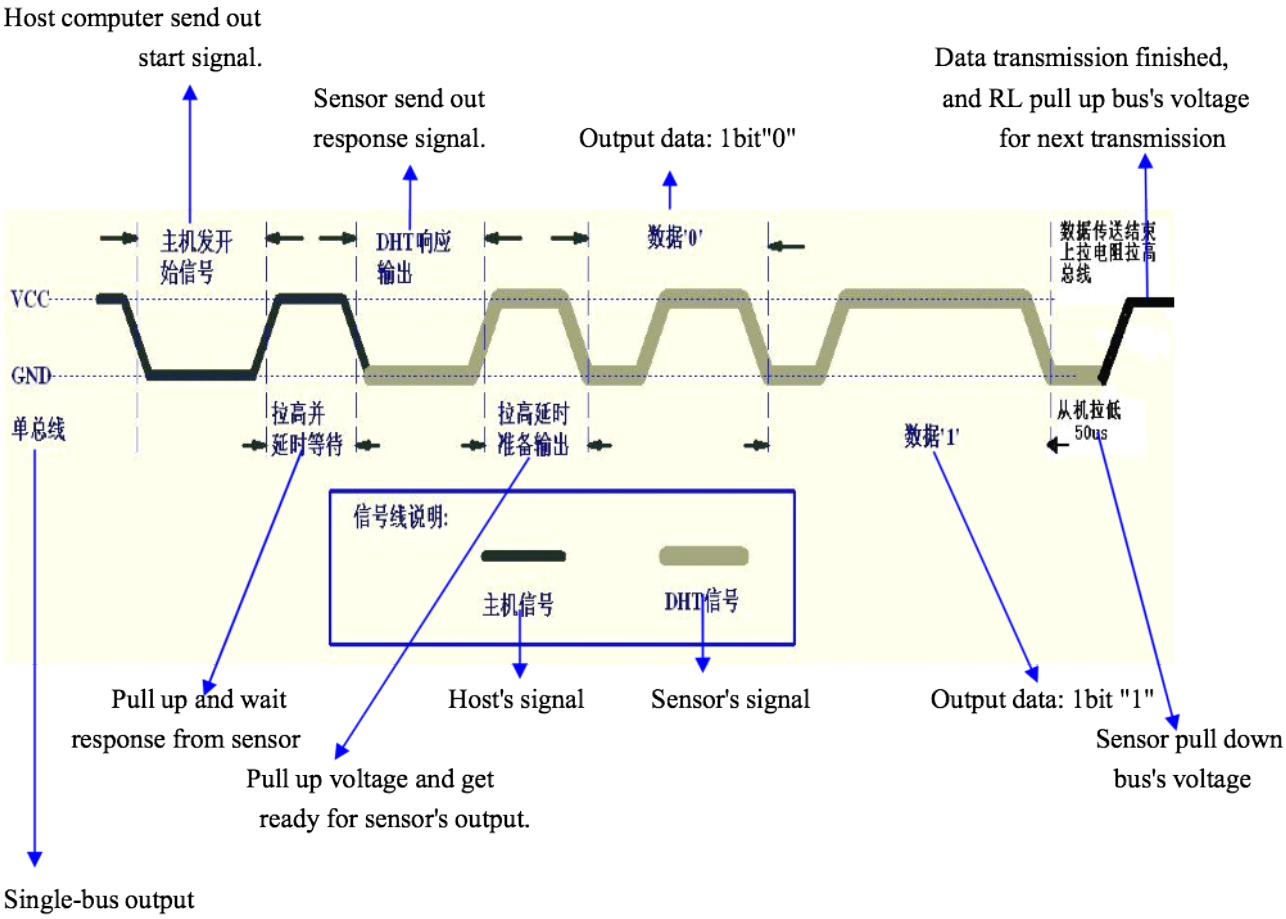


Figure 4: Overall transaction communication

Additional Comments

To capture the sensor data bits, RPi can use either poling or interrupt mechanisms. Polling requires reading and capturing data bits at specific time sequences which are provided by the DHT11 specs. So RPi polling could be very sensitive to the accuracy of these timings especially at the microsecond level.

Interrupts can respond to rising or falling edges of the time pulses, thus are less sensitive to the precise timings based on the specs.

The wiringPi library provides special functions for RPi to issue microsecond delays as well as respond to edge trigger events. Details and syntax can be found in the wiringPi documentation site.