Lab20: I2C Communications

Chengming Li

10/26/2022

1. Background

The Inter-IC bus is a common digital bus. There are really four pins associated with the I2C bus: SDA, SCL, GND, Vdd.

The SCL is unidirectional transport and has the clock signal that travels only in one direction.

The SDA is bidirectional. Each device on the I2C bus is connected in parallel to all the other devices and the controller. Each minion device has an address that it listens for on the data line.

The SCL and SDA drivers for all minions and some controllers are “open collector” or “open drain” This means the drivers are connected to the bus by an open collector. This means when the driver wants to write a 0-bit, it turns the transistor on, pulling the line down.

But it has no way of writing a 1-bit, or a HIGH signal. Instead, it relies on a pull up resistor to pull the line HIGH when the transistor is turned off.

The resistor value used in design: A large R means the pull-up time constant gets longer but lower power dissipation. A small R means the pull—up time constant << the unit interval of a bit, but more power dissipation.

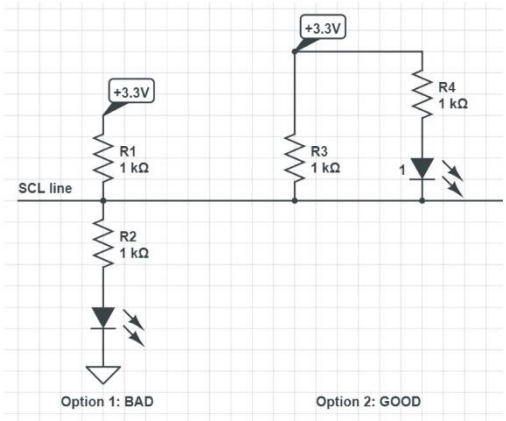
2. Measurement of I2C Bus

By measuring the SDA or SCL lines while the Arduino is driving I2C communications, I loop at the impact on the rise time of adding a 1k and 10k pull resistor. The probes are measured at the SCL and SDA lines.

The left picture shows the rise time of SCL(blue trace) with **1k** pull-up resistor. And the rise time is around 100ns. The right picture shows the rise time of SDA(yellow trace), which is around 110 ns. These two values are close to each other.

The left picture shows the rise time of SCL(blue trace) with **10k** pull-up resistor. And the rise time is around 470ns. The right picture shows the rise time of SDA(yellow trace), which is around 455 ns. These two values are close to each other.

As shown above, a 1k pull-up resistor would be a great choice for designing the I2C bus in the Arduino Board, because it satisfies the time requirement of charging C and sending data.



Beyond that, if we want to connect LEDs to the SCL and SDA lines. Option 2 is better than 1. The reason is the first option has pull-up resistors connected in series, which create a voltage divider in the SCL line. And only 1.65V will be read in the SCL line, which is not enough to write a HIGH signal. And, from google, the minimum voltage required to have a signal read as HIGH is 2V. In the first case, it is not enough to drive the signal HIGH. But in the second implementation, resistors are connected in parallel, voltage drops are the same in both resistors, which means there is enough voltage to drive SCL HIGH.

The Thevenin resistance inside the Arduino in my case is around 12.5K. It is calculated by modeling the pin as a Thevenin source. The open circuit voltage is the Thevenin voltage, and the loaded voltage is related to the external resistor and the Thevenin resistance, 10k in my case.
In SCL line:
Vload = Vopen * (R_intern/(R_intern+Rload)) -> 2.61 = 4.75 * (R_intern/R_intern+10k) -> R_intern = 12.2k.

3. Recommendation for the pull-up resistor

If the internal capacitance is 30p F, based on the formula:
2.2*RC < 0.1*Operating Frequency.

For 100kHZ:
2.2*R*30p < 0.1* 100us -> R < 15.1k

For 400kHz:
2.2*R*30p < 0.1*2.5us -> R < 4K

If there are multiple devices connected to the I2C, how many pull-up resistors should I add?
The I2C bus only needs one set of pull-up resistors, regardless of how many target devices are connected.