

APPLICATION NOTE

AN452

One mile long I²C communication
using the P82B715

Author: Don Sherman

October 12, 1994

Philips Semiconductors



PHILIPS

One mile long I²C communication using the P82B715

AN452

Author: Don Sherman, Applications Engineer

LONG DISTANCE I²C DESIGN CONSIDERATIONS

Summary

I²C communication is possible over very long distances, but at greatly reduced data rates. Tests were made and results shown for cable lengths exceeding one mile. This report explores the design considerations and trade-off needed to implement a successful long distance I²C communication system.

In Application Note AN444 I described how the introduction of the I²C extender chip, the Philips P82B715, allowed I²C to be taken out of the TV set environment and be used to interconnect many nodes, separated by hundreds of feet of distance, and still allow operation at the full 100Kbps data rate. The large addressing range of the I²C protocol makes the large number of stations possible and the P82B715 enables the long distance communication to take place. There are, obviously, limitations and constraints which must be observed, but the ability to take an existing I²C system and to now connect it to other systems without the need for intermediate protocol translators, different power supplies, and expensive interface chips, can be very useful. Needless to say, the user, whose confidence is bolstered by the newly expanded boundaries, wants to know the new rules and the new limits so that he can plan. At 100Kbps the limiting effects eventually boil down to the time that it takes to charge up the cable capacitance through the pull-up resistors. Since the I²C is a two-wire system, there is no common mode noise rejection, so it is inferior to RS422, RS423 and RS485 systems which do use differential drivers and receivers. Since the I²C protocol requires that the Master generate the clock signal which is used by the Slave, it is not tolerant of long phase delays. The Slave must receive the clock and output its data and this data must be received by the Master before the Master starts the next clock cycle. So I²C is inferior to RS232 which, because of its self-clocking, can actually tolerate minutes of phase delay and still deliver the

message at 115 Kbps. Thus, I²C requires at least some form of shielded cable because of its poor tolerance of noise injection, and this shielded cable brings with it the capacitance which ultimately limits the slew rate of the signal.

AN444 showed that shielded audio cable could be used to communicate up to 1000 feet at a 100Kbps data rate. What if the user wants to go farther? The answer is that you can, but you have to do it at a slower data rate. In this Application Note tests were run up to a distance of 6,000 feet, and the effect on the data rate was noted. The results are shown in Figure 1. The tests were run with the I²C hardware on the 87C751. Since there is not a great deal of flexibility in the timing from Timer 1, the only way to reduce the data rate is to reduce the crystal frequency or use software "bit banging". This Application Note used the on-board I²C with the reduced crystal frequency technique. The results are shown in Figure 2.

The reduction in the I²C data communication rate for long distances, produces a dilemma for the user. Is the advantage of being able to directly address every user in the loop worth the trade-off of overall communication (and processing) speed reduction within the local nodes? For those who need the full 100Kbps speed in the local node, it will be necessary to use some form of mail box to store messages and then some other method will be used to forward these mail packets to the far away nodes. This secondary method could be I²C, but it will require a completely separate system from the high speed one. If I²C is chosen, the slow version should probably be just "bit banged". One such program for doing this is available on the Philips Microcontroller BBS entitled I2CBITS.ZIP. There are also several programs on the BBS to "bit bang" RS232, if a UART is not available.

The rest of this Application Note is devoted to the user who can tolerate the overall system speed reduction and wants to use the slow I²C protocol throughout his system.

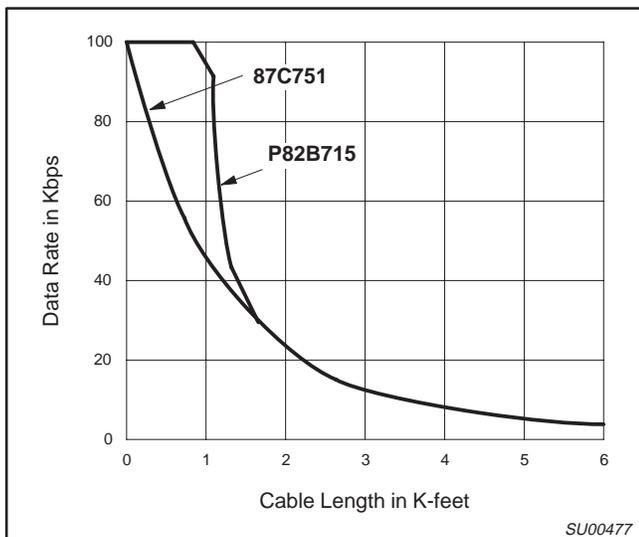


Figure 1.

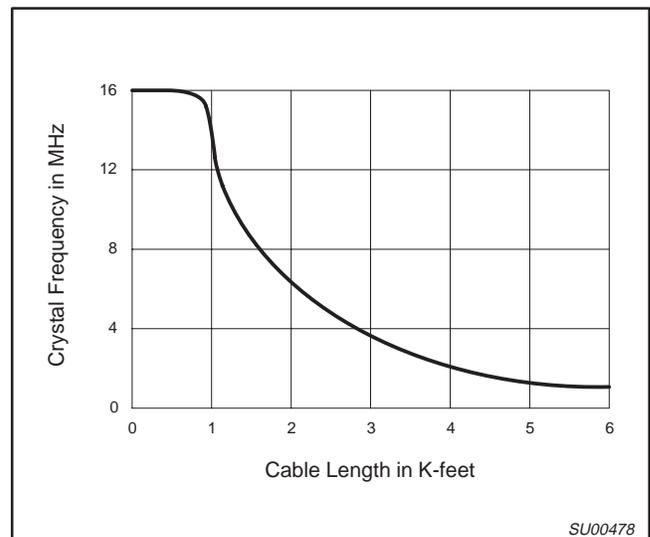


Figure 2.

One mile long I²C communication using the P82B715

AN452

Table 1. Worst case model, 4.0 volts at 130°C

I _{OL} (mA)	V _{OL} = 0.3V	V _{OL} = 0.45V	V _{OL} = 0.6V	V _{OL} = 1.0V
87C751	9.0	12.4	15.1	18.9
83C752	7.3	10.4	13.3	19.7
87C654	9.1	12.5	15.1	18.5

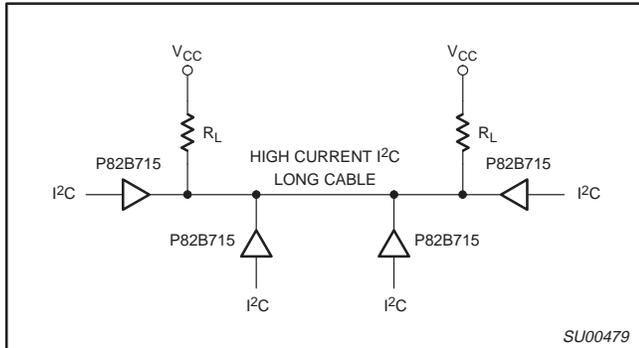


Figure 3.

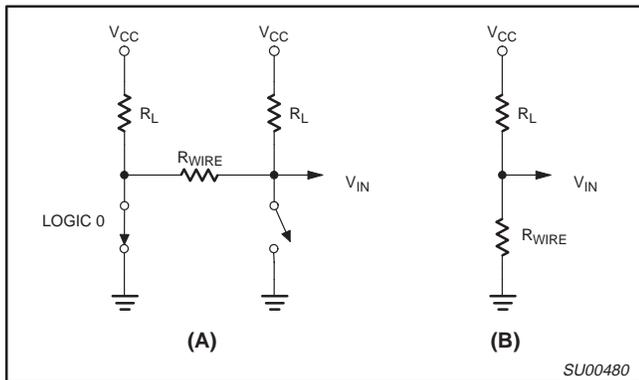


Figure 4.

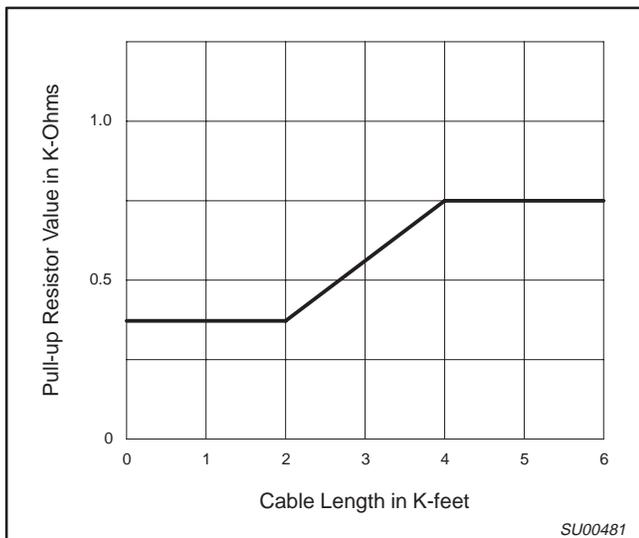


Figure 5.

The high speed, long distance I²C system was made possible by the introduction of the P82B715 I²C extender chip. This IC consists of two identical X10 current amplifiers. The overall system is shown in Figure 3. The value of the pull-up resistor, R_L, determines the rise time of the signals. Current flows through R_L to charge up the capacitance of the long cable. This requires that a non-standard, intermediate, I²C bus be created which must be de-buffered with another P82B715 at the other end of the line. The local I²C nodes require a 3 mA pull-up current, and the buffered I²C line uses a 30 mA total pull-up current. Note that there are pull-up resistors at each end of the buffered I²C line, so that each end will have drive capability.

As the cable length becomes longer than 1000 feet, the effect of the cable wire resistance starts to become a limiting factor. Figure 4(A) shows a typical system and Figure 4(B) represents the case of detecting a remote logic low signal. Figure 4(B) shows that eventually the resistance of the wire will drop so much voltage that the minimum V_{IN} will be greater than the maximum allowable V_{IL} of the receiving buffer. Thus, it is necessary to increase the value of R_L to keep V_{IN} within reasonable bounds. Note that the wire resistance is fixed and determined by the AWG of the wire and the length. The increase in R_L reduces the available charging current, while at the same time, the increased wire length proportionally increases the capacitive load which must be charged. The propagation time for the signal to go down and return also increases with cable length. This multi-fold effect produces the non-linear curve shown earlier in Figure 1. The increase in load resistance required for the buffered system is shown in Figure 5.

The forced increase in the allowable value of the pull-up resistor, R_L, reduces the need for a powerful bus extender. Experiments were conducted where the microcontroller's I²C outputs were directly connected to the long distance cable. The model for the voltage divider is shown in Figure 6. Note that the R_{DS(ON)} of the output NMOS transistor must sink the sum of all of the pull-up currents, whereas the voltage drop in the cable is only from one pull-up resistor. The values of the pull-up resistor driven directly by the 87C751 microcontroller are shown in Figure 7. The interesting conclusion of testing these two versions, with and without the P82B715 bus extender, was a communication rate vs. distance graph which was virtually identical, thus Figure 1 holds for both methods. The reason for the good performance of the Philips Semiconductors 87C751 microcontroller in these experiments can be seen by examining Table 1.

The tests for this Application Note were run using 1000 foot coils of shielded Belden miniature sound and control Cable, Belden 9452, each roll which has a ground wire of 19.0Ω and a twisted pair of 24AWG wires each with 25.4Ω resistance/1000'. The measured capacitance was signal to ground of 60nF/1K' and with a signal to signal wire capacitance of 32nF/1K'. All of the nodes were powered by floating, transformer isolated power supplies, with only one tied to ground. Large voltage spikes are generated in the ground wire when the outputs are driven to ground. The effect is shown in Figure 8 and discussed later in this Application Note.

One mile long I²C communication using the P82B715

AN452

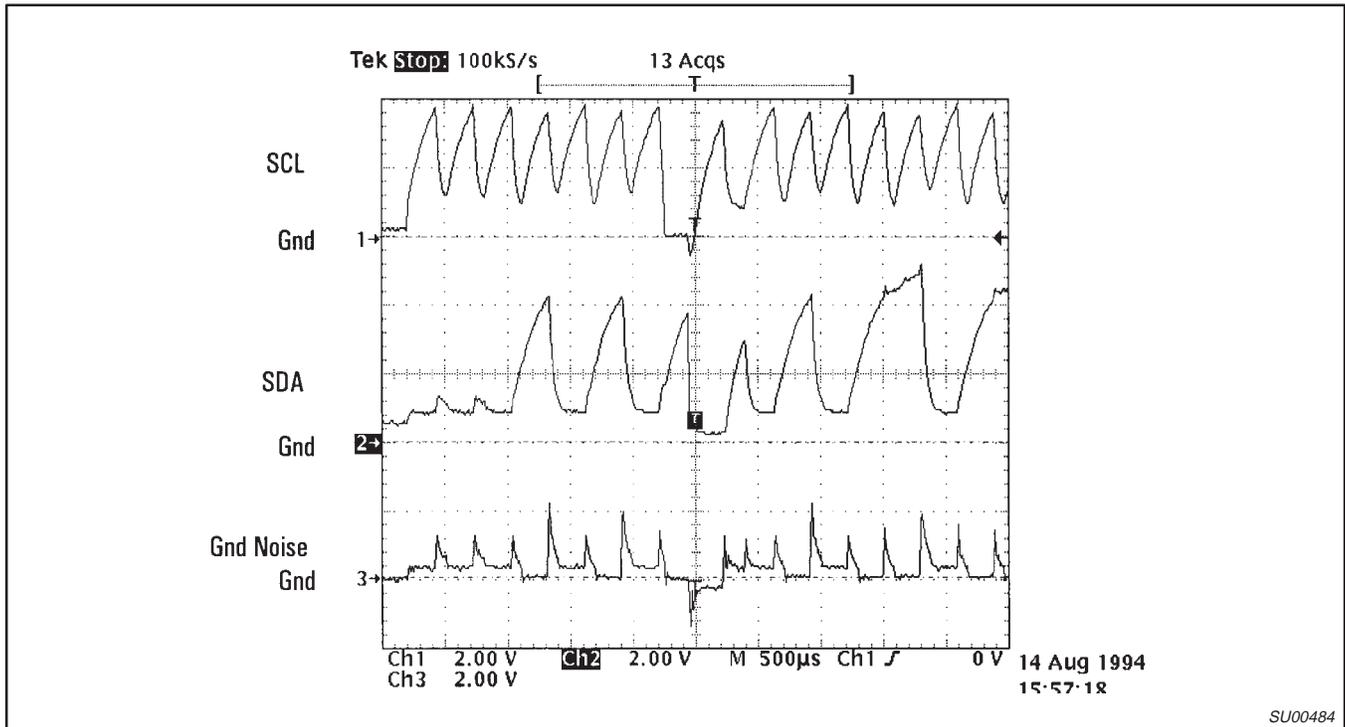


Figure 8.

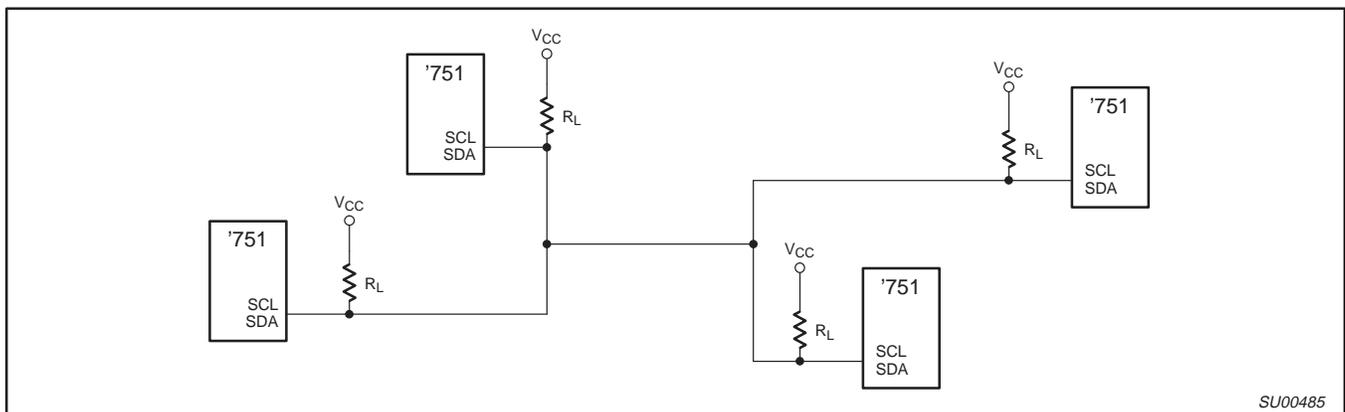


Figure 9.

One mile long I²C communication using the P82B715

AN452

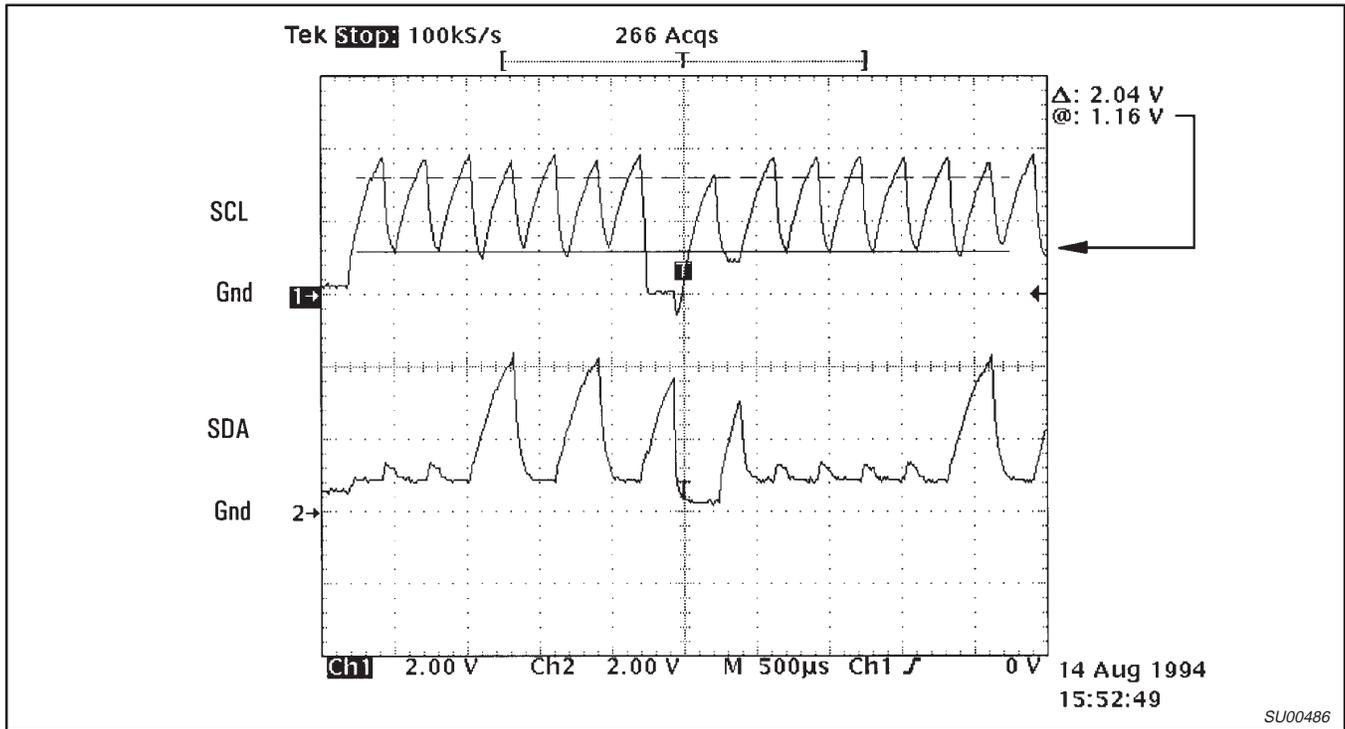


Figure 10.

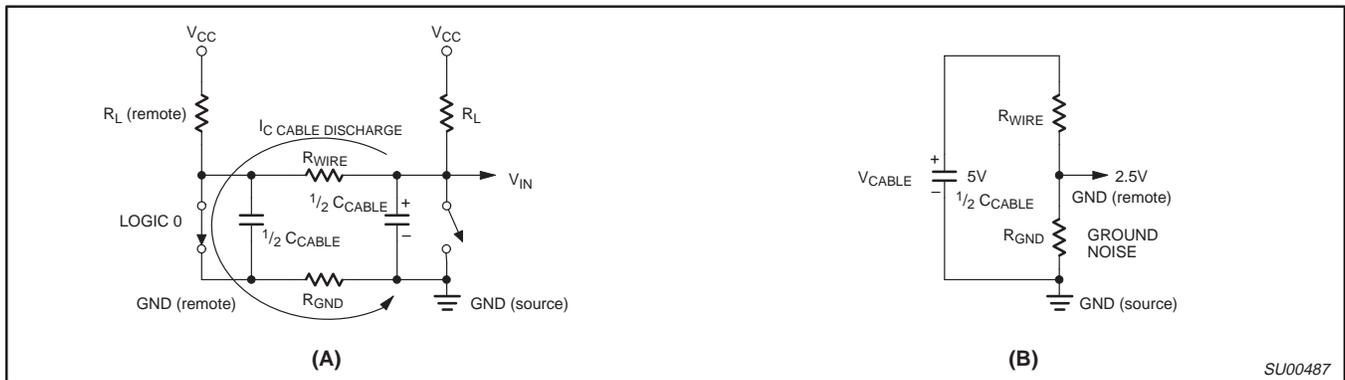


Figure 11.

One mile long I²C communication using the P82B715

AN452

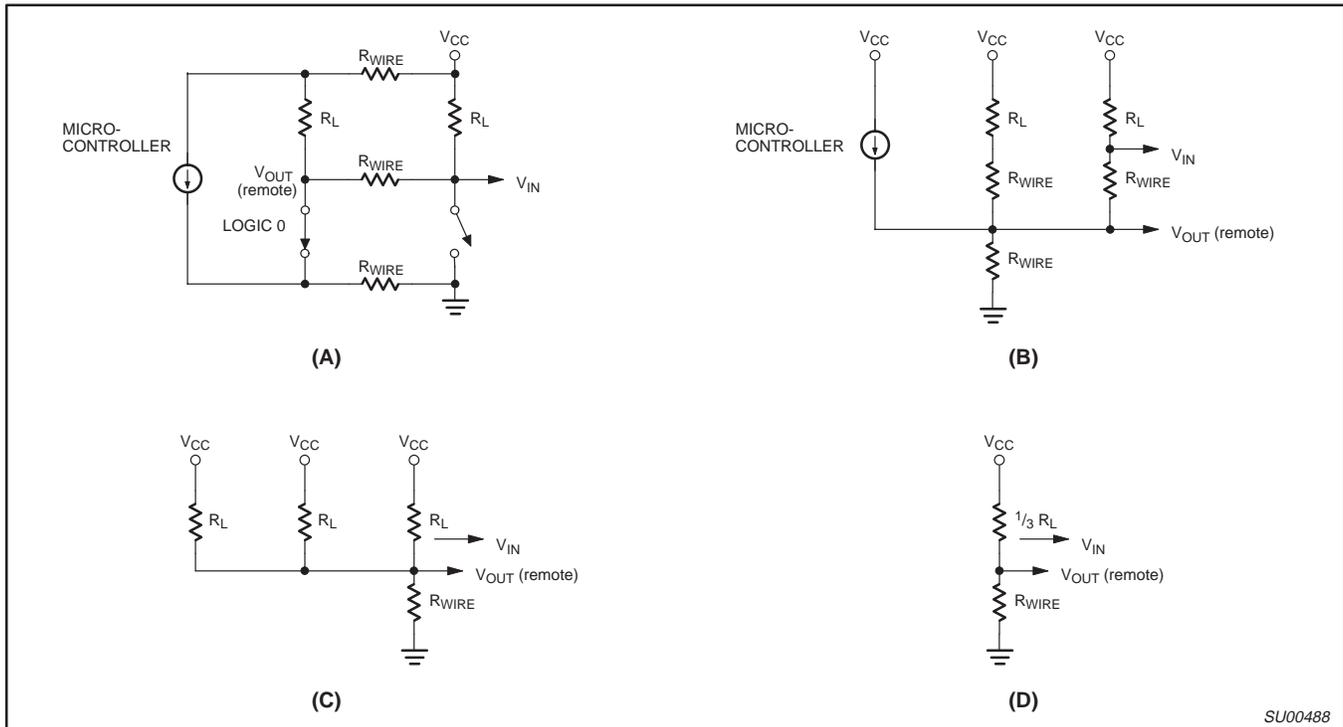


Figure 12.

The effect of the voltage drop in the wire can be seen in Figure 10. Since the microcontroller has TTL translators on the inputs, the actual crossover point is around 1.2 volts. Notice that there is no noise margin left in the scope picture, as the low levels are virtually at the crossover point of the microcontroller's input. Coupling this low, to almost non-existent, noise margin with the horrible ground noise spikes, shown in Figure 8, make the more cautious engineer wonder how the thing works at all.

The noise spikes can be better understood by looking at the AC model for the network as shown in Figure 11. When the remote unit switches to ground, the transition occurs quite rapidly compared to the slow rise time. The closest capacitance (1/2 C_{CABLE}) is quickly discharged by the switch, whereas the receiving side's cable capacitance must complete the current discharge path shown in Figure 11(A). Notice that from the simplified model, Figure 11(B), the instantaneous voltage between the source ground and the remote ground will be 2.5 volts (1/2 V_{CC}) if the cable had been charged all of the way to the 5 volt rail before being driven to ground by the remote unit. The Ground Noise shown on trace 3 of Figure 8 is the difference between the two grounds.

Sending Power Through the Cable

It would be desirable if the remote units in the system could receive their power through the cable. Figure 12(A) shows a model for such a system. The current source is a representation of the current actually consumed by the microprocessor while it is sending a logic 0 to the power sourced unit's V_{IN} pin. The problems with voltage drop in the V_{CC} line can be overcome by sending a higher power supply voltage than needed and then using a voltage regulator at the remote micro's location to provide the regulated local power. The voltage drop in the ground line is the most critical and the model in Figure 12(B) is modified to simplify the ground voltage drop calculations. This figure transposes R_L and R_{WIRE} and ignores the V_{CC} supply drop. Figure 12(C) further simplifies the circuit since R_L is usually 10 times R_{WIRE}. It also replaces the

approximately 5mA current drawn by the microcontroller at the lowered clock rate, required for the slow I²C configuration, with a resistor equal to the load resistor, i.e., 5mA at 5 volts is equivalent to a 1KΩ resistor and this happens to be the value of the R_L used in this example. Finally, Figure 12(D) shows a very simplified model for ground wire and R_L calculations. V_{OUT}(remote), the output voltage at the remote sender, will be the largest contributor to the V_{IN}. For R_L = 10 R_{WIRE}, V_{OUT} becomes approximately 1/4 of the V_{CC} supply voltage.

$$\begin{aligned} V_{out} &= V_{cc} [R_{wire}/(R_{wire} + 1/3R_L)] \\ &= V_{cc} [R_{wire}/((R_{wire} + 1/3(10R_{wire})))] \\ &= V_{cc} [1/(1+10/3)] \\ &= V_{cc} [1/(13/3)] \\ &= 5V (3/13) \\ V_{out} &= 1.15Volts \end{aligned}$$

From Figure 12(B), V_{IN} is about 1/10 of the supply voltage, (0.1 V_{CC}). Thus, from Figure 12(D), V_{IN} = V_{OUT} + 0.1V_{CC}. For the case described above, this becomes:

$$\begin{aligned} V_{in} &= 1/4 V_{cc} + 1/10 V_{cc} \\ &= V_{cc} (2.5 + 1)/10 \\ &= 5V 3.5/10 \\ &= 0.35(5V) \\ V_{in} &= 1.75 volts \end{aligned}$$

Since the crossover from a logic 0 to a logic 1 is at around 1.2 volts, it is obvious that even one remote unit cannot be supported with this cable for R_L = 10 R_{WIRE}.

Conclusion

The additional voltage drop caused by running DC power through the common ground line only serves to reduce the already small noise margin. To compensate for this, the user must either reduce the AWG of the wire to provide a lower IR drop, and/or increase the value of the pull-up resistor, which will slow down the communication rate.

One mile long I²C communication using the P82B715

AN452

Another possibility for providing power down the cable to the remote units consists of alternately charging a capacitor at each node and then using the current stored there to temporarily power the slave micro for a long enough time to get data out of it (see Figure 13). Thus, no communication is attempted during the capacitor charge cycle, so the high voltage drop along the ground line has no deleterious consequences. During the playback time, the individual nodes will have their own local power supplied through its storage capacitor, so that no ground current is needed to power the microprocessor.

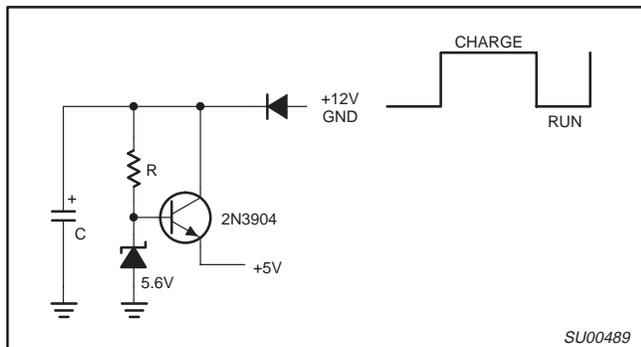


Figure 13.

As an example, twelve volts would be sent down a fourth wire, the "power line", for a sufficient amount of time to charge all of the capacitors. Each capacitor is connected to the power line through a diode. Once the capacitors are fully charged, the power line is taken to ground. The master would then have to interrogate each of the slaves before their power capacitor dropped below the dropout voltage of its regulator. Let us consider two possible scenarios:

In the first example, five remote slaves will be distributed along a 2000 foot cable. From Figure 1, the maximum data rate is 20Kbps. If each slave delivered 2 bytes of response to its one byte address, the total number of bits of data passed to talk to all five units would be:

$$5 \text{ units} \times 3 \text{ bytes of information} \times 8 \text{ bits/byte} = 120 \text{ bits}$$

$$\text{Time to transmit 120 bits} = 120\text{bits}/20\text{Kbits/sec} = 0.006 \text{ sec}$$

Thus, the power supply capacitor must be able to provide current for at least six milli-seconds. To be safe, let's assume a factor of three margin. This results in a time constant of 18mSec. Assuming that the time to discharge the capacitor to the dropout voltage of the regulator is one RC, and that the current per microcontroller plus the SDA pull-up resistor current (remember that the master drives SCL) will be about:

$$10\text{mA micro } I_{CC} + 5V / 0.5\text{k}\Omega = 10\text{mA} + 10\text{mA} = 20\text{mA.}$$

The 20mA load can be approximated by a resistor which will deliver 20mA at the five volt V_{CC} supply:

$$R = V_{CC}/I_{LOAD} = 5V/20\text{mA} = 250\Omega$$

$$RC = 18\text{mSec}$$

$$C = 0.018\text{sec}/250\Omega = 72 \mu\text{F}$$

Let's use 100μF to give us some elbow room. Each node will need a 100μF capacitor to provide power for 18mSec. The worst case charging time would be if all of the units were at the full 2000' distance. To insure 99% charging, three RC time constants will be used. The R is the resistance of 2000' of cable:

$$RC = (2 \times 25\Omega/1\text{K}') \times (5 \text{ nodes} \times 100\mu\text{F}) = 0.025 \text{ sec}$$

Thus, the master can interrogate each node for two bytes of data every 31 mSec. Twenty five mSec will be used to pump up the storage capacitor and it then takes six milli-seconds to transfer 120 bits of information at 20Kbps.

The second example would be a 64 remote unit system driven by one master over a 4000 foot cable. If each slave delivered 2 bytes of response to its one byte address, the total number of bits of data passed to talk to all 64 units would be:

$$64 \text{ units} \times 3 \text{ bytes of information} \times 8 \text{ bits/byte} = 1536 \text{ bits}$$

At a data rate of 3Kbps, this would take about 1/2 second. Thus, the power supply capacitor must be able to provide current for around one second to be safe. Assuming that the time to discharge the capacitor is one RC, and that the current per microcontroller plus the SDA pull-up resistor current will be about 5 mA, this load can be approximated by a 1 KΩ resistor:

$$RC = 1 \text{ second}$$

$$C = 1/R = 1/1000\Omega\text{s} = 1000\mu\text{F}$$

This is probably too small, but for the sake of calculating, lets use it. Now the total number of capacitors to be charged is equal to the number of nodes, which in this case is 64. So, the time to charge 64 × 1000μF will be determined by the resistance of the wire that the current is flowing through. Four thousand feet of 24AWG wire has about a 100Ω of resistance, therefore the time to charge up all of the capacitors will be approximately:

$$\begin{aligned} \text{Time to charge to 98\%} &= 3 RC = 3 \times 100\Omega \times 64,000 \mu\text{F} \\ &= 19,200,000 \text{ micro seconds} \\ &= 19.2 \text{ SECONDS} \end{aligned}$$

Now if additional capacitance is added for some elbow room, this system could only deliver 2 bytes of data from each unit every 30 seconds.

Note that only the units at the far end of the cable will see the full 100Ω of wire resistance. If the master is told which units are farthest away, it could shorten the charging cycle, realizing that the farthest units are not fully charged. If they are addressed first, it might be possible to shorten the total capacitor charge cycle time.

CONCLUSION

By using a local storage capacitor at each node, it is possible to alternately charge the capacitor and then use this charge to power the remote units long enough for the master to interrogate each of the remotes. This system eliminates the problems caused by the reduction in voltage margin caused by passing the microcontrollers' I_{CC} through the signal ground line.

One mile long I²C communication using the P82B715

AN452

NOTES

One mile long I²C communication using the P82B715

AN452

Philips Semiconductors and Philips Electronics North America Corporation reserve the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

LIFE SUPPORT APPLICATIONS

Philips Semiconductors and Philips Electronics North America Corporation Products are not designed for use in life support appliances, devices, or systems where malfunction of a Philips Semiconductors and Philips Electronics North America Corporation Product can reasonably be expected to result in a personal injury. Philips Semiconductors and Philips Electronics North America Corporation customers using or selling Philips Semiconductors and Philips Electronics North America Corporation Products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors and Philips Electronics North America Corporation for any damages resulting from such improper use or sale.

Philips Semiconductors
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088-3409
Telephone 800-234-7381

Philips Semiconductors and Philips Electronics North America Corporation
register eligible circuits under the Semiconductor Chip Protection Act.
© Copyright Philips Electronics North America Corporation 1994
All rights reserved. Printed in U.S.A.