I^2C Maximum Clock Speed Calculator

Mar 2009

BLSIC/BU MMS
NXP semiconductors

I^{2}C-Bus Maximum Clock Speed Calculator

Excel file of Calculator >

Bus Speed
Calculator

Double click to open.

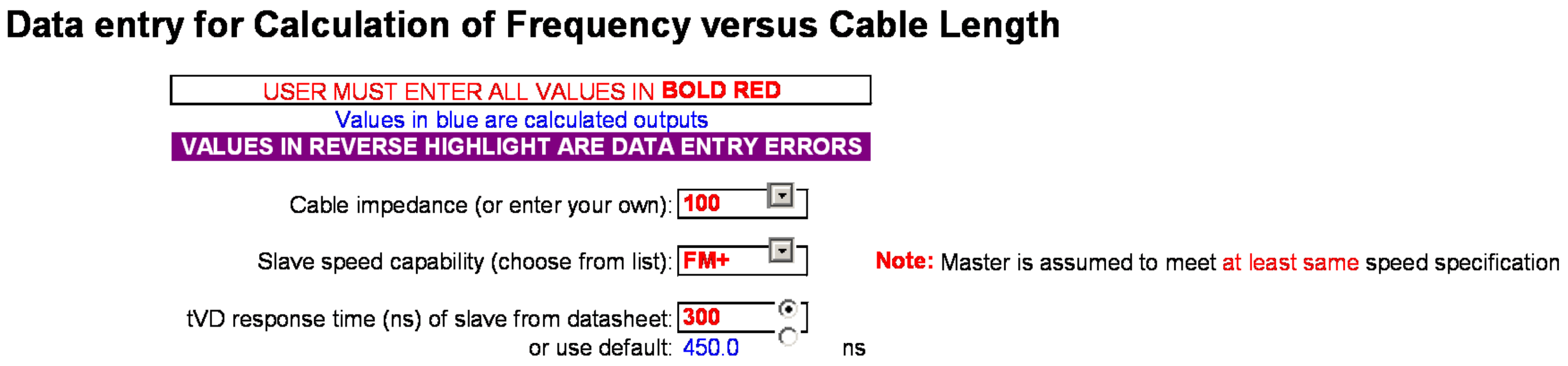
If this is a PDF then go to URL

http://www.standardics.nxp.com/support/documents/i2c/?type=software

for the Excel file labeled:

I^{2}C Buffers: Calculation of Bus Frequency versus Cable Length (Oct 2007)

Data entry sheet



Data entry for Calculation of Frequency versus Cable Length

USER MUST ENTER ALL VALUES IN **BOLD RED**

Values in blue are calculated outputs

VALUES IN REVERSE HIGHLIGHT ARE DATA ENTRY ERRORS

Cable impedance (or enter your own): **100**

Slave speed capability (choose from list): **FM+**

Note: Master is assumed to meet at least same speed specification

tVD response time (ns) of slave from datasheet: **300** ●

or use default: 450.0 ○ ns

Step 1: Open the calculator and select the Data Entry sheet.

The user will be asked to enter all data that is shown in **bold red** font

It is not necessary to enter all values immediately.

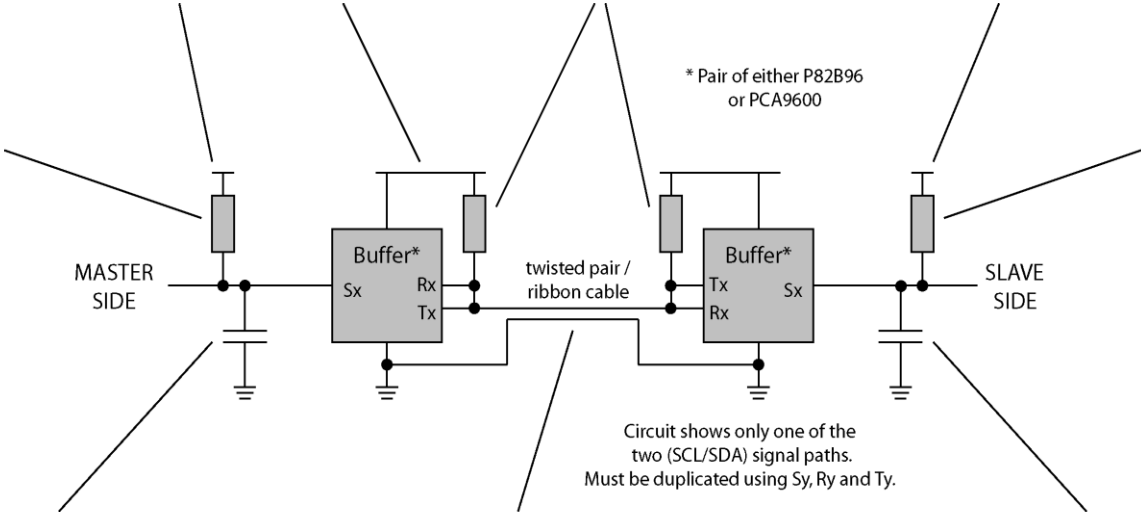
First enter the values that are known, for example select the type of cable that will be used, or enter the characteristic impedance of special cables that will be used and are not listed.

Select the I^2C speed capability of the (slowest) slave that is included in the system e.g. Fm+ or Fast Mode etc.

(The calculations assume that the Master has a speed capability at least equal to the entered slave speed. When the Master and slave have very different speed classes it is VALID to enter a fast slave timing E.G. Fm+ but the user must enter appropriate bus Hi/Lo periods for a slower Master as these may NOT be correctly prompted, see slide 6)

Enter the tVD response time of the (slowest) slave or, if no data is published, select to use the default and the calculator will use the maximum response time for devices of the I^2C bus class selected.

Data entry sheet (continued)



Bus voltage at master side
3.3 V

Cable bus voltage
5.0 V

Pull-up resistors on
Tx/Rx at each end of cable
Minimum 333.3 ohms
Actual 317.0 ohms

Bus voltage at slave side
3.3 V

Pull-up resistor on master bus

P82B96	PCA9600	
966.7	1450.0	ohms Minimum
1000.0	1000.0	ohms Actual

Pull-up resistor on slave bus

PCA9600	P82B96	
966.7	966.7	ohms
1000.0	900.0	ohms

50.0 pF
Capacitance on bus at bus master

(Cable lengths
for graph)
Longest 50.0 m
Shortest 0.0 m

My cable's actual length 30 m

50.0 pF
Capacitance on bus between
PCA9600 and slave (LED driver)

Step 2: Enter all the supply voltages that will be used.

The calculator will respond by calculating the minimum values that are allowed for the pull-up resistors and these will be displayed in blue font.

If the total capacitance on the Master and slave buses is known, enter those values. (It is generally not so important but where it will be larger than, say, 50pF it is best to enter an approximate value. 10pF per connected chip will usually be a reasonable estimate)

Enter the pull-up resistors that will be used. They should be larger than the minimum calculated value

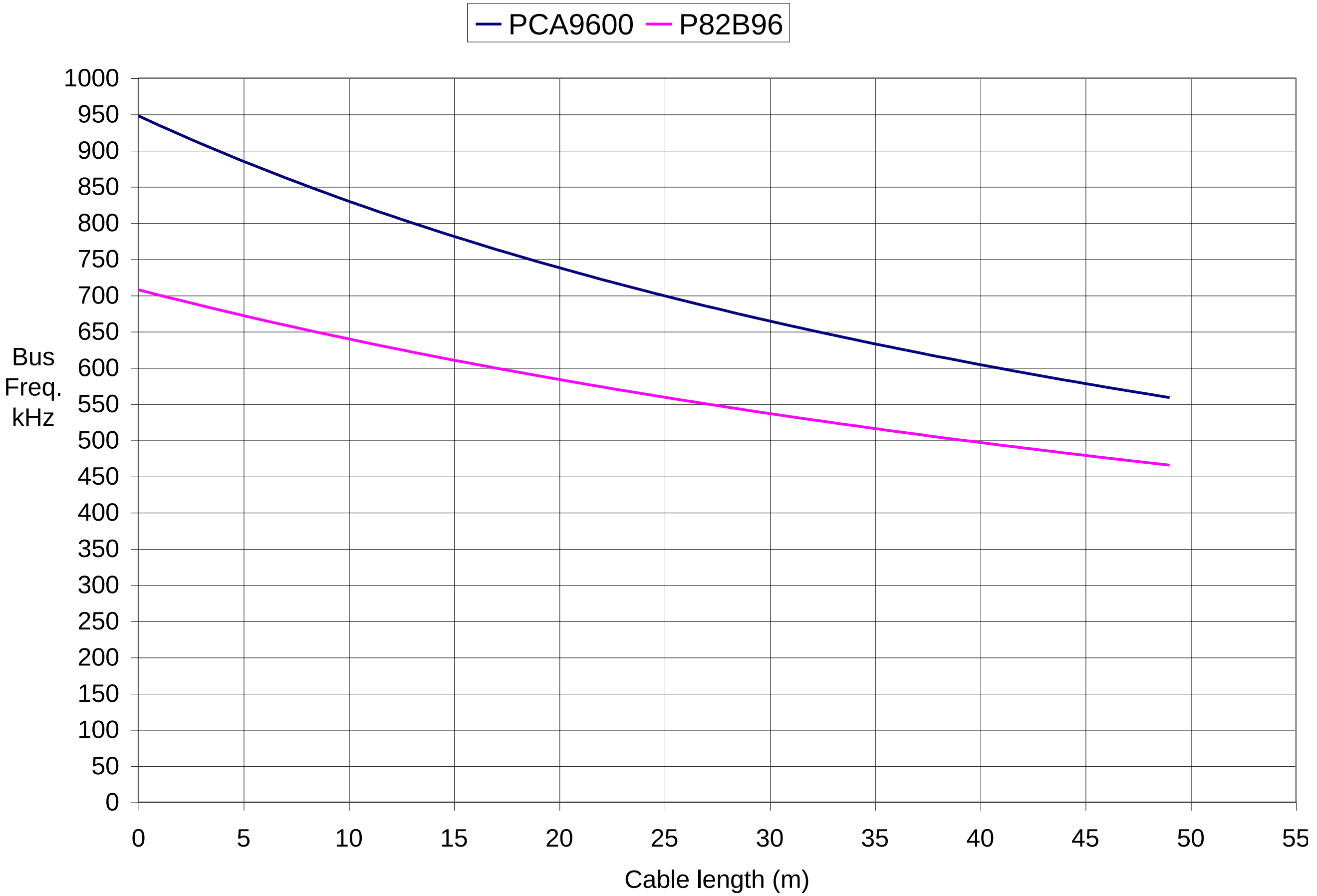
Enter the range of cable lengths of interest. The graph will show bus speeds for the range selected.

Chart sheet

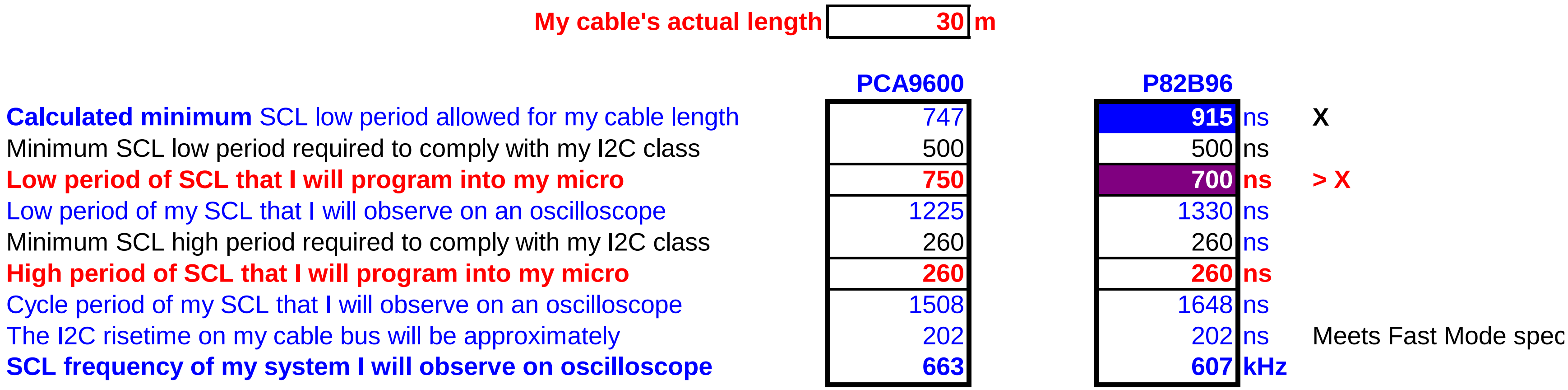
Step 3: Select the chart sheet.

The calculator has drawn a graph showing the maximum (actual) bus frequency against cable length for P82B96 and PCA9600. “Actual” means the bus frequency that would be observed on an oscilloscope.

The chart assumes the Master’s High and Low periods of SCL can be programmed to the ideal values.



Data entry sheet (re-visited)



My cable's actual length 30 m

	PCA9600	P82B96		
Calculated minimum SCL low period allowed for my cable length	747	**915**	ns	X
Minimum SCL low period required to comply with my I2C class	500	500	ns	
Low period of SCL that I will program into my micro	**750**	**700**	**ns**	> X
Low period of my SCL that I will observe on an oscilloscope	1225	1330	ns	
Minimum SCL high period required to comply with my I2C class	260	260	ns	
High period of SCL that I will program into my micro	**260**	**260**	**ns**	
Cycle period of my SCL that I will observe on an oscilloscope	1508	1648	ns	
The I2C risetime on my cable bus will be approximately	202	202	ns	Meets Fast Mode spec
SCL frequency of my system I will observe on oscilloscope	**663**	**607**	**kHz**	

Step 4: Return to the Data Entry sheet and refer to the data table as displayed above.

Enter the cable length (in meters) that will be actually used at "**My cable's actual length**"
The calculator responds by calculating the minimum allowed SCL low period.
(It also lists the minimum allowed SCL low period for the I^{2}C class that was entered as a reminder of this additional requirement.)

Enter the low period of SCL that will actually be programmed into the Master device.
It should be longer than the calculated minimum AND longer than the specified minimum for the I^{2}C speed class selected. If these tests are not passed the user entered value is highlighted and a reason shown. The requirement for P82B96 is marked as "X" and the entered value is prompted to be made ">X"

Enter the high period of SCL that will actually be programmed into the Master device. It should exceed the minimum shown

The calculator outputs (just for information because there is no requirement to meet any target) the nominal rise time on the cable bus

The calculator outputs the actual SCL cycle period and frequency, as will be observed on an oscilloscope, for the particular system parameters that have been entered by the user.

(Because some 'worst case' timings are used it will not be unusual to sometimes observe that the bus actually runs at a higher frequency than calculated
If the Master has no possibility to enter separate values for the SCL high and SCL low then the SCL high to enter must be made the same as for the low.)

Details of the calculations

The following slides show the factors that are being calculated by the spreadsheet in order to predict the system speed. It is necessary to make some assumptions, and restrictions, in order to make reasonably accurate predictions.

The calculations are based on a simple bus system with just two nodes.

All Masters were assumed to be at one end of the cable and the slaves at the other end.
(Because the cable link including the buffers is symmetrical it is possible mix Masters/slaves at each node PROVIDED the Masters have, as a minimum, the same timing parameters such as tVD as are entered for the slave and the bus supplies/pull-ups/capacitance at each node are made reasonably similar.)

The cable bus is buffered using either P82B96 or PCA9600 at each end so their characteristics have been used.

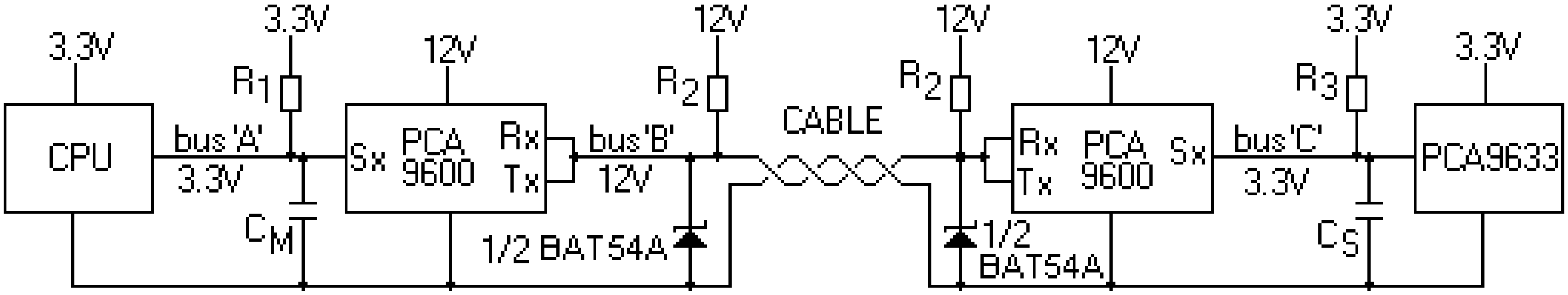
The cable bus terminations (pull-ups) are symmetrical, i.e. they are the same value at each end of the cable.

Schottky diode clamps (BAT54A or similar) have been fitted between the Tx/Rx signal pins and the IC ground pin at both ends of the cable to absorb the transient energy in the cable and to limit the negative voltages on those IC pins.
(For clarity of identifying signals, these diodes are not shown on EVERY slide in which partial schematics are used as illustrations, and they are not shown on the Data Entry schematic but in **all** cases they are **assumed** to have been fitted.)

(In the calculator, the sheets labelled PCA9600 and P82B96 perform the timing calculations and may provide useful details for very experienced system designers. For example systems with multiple nodes e.g. systems with Fm+ devices directly connected along the cable are complex to analyse, but some partial application of this spreadsheet calculator or application of its details may be helpful. The last slide gives some general guidance regarding one approach to designing multiple-drop systems (with or without slave buffering) but it cannot cover the many variables.

Timing in a system with PCA9600 (or P82B96) buffers

Typical system using PCA9600 to drive long wiring with 12V logic signals



The Master CPU generates SCL timing of High and Low periods

The bus capacitance and pull-up resistor on this bus determine the rise and fall times of this Master bus

PCA9600 changes the 3.3V CPU signal levels to 12V cable bus signals.

The I²C signal will be delayed whenever it passes through this buffer chip.

The pull-up resistors on the cable bus determine the cable bus rise time.

The cable length determines the signal delay.

The cable's characteristic impedance plus the drive current of PCA9600 determines the bus fall time

Schottky diode clamps limit negative signals (overshoot)

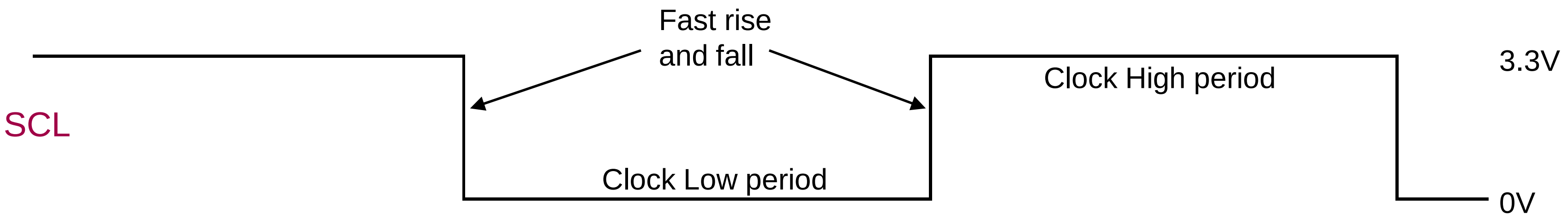
PCA9600 changes the 3.3V CPU signal levels to 12V cable bus signals.

The I²C signal will be delayed whenever it passes through this buffer chip.

The bus capacitance and pull-up resistor on this bus determine the rise and fall times of this Slave bus

The slave receives the Master clock and data and generates the Acknowledge signal or data signals required by the Master CPU

Master generates SCL timing of High and Low periods



If there was no buffer connected to the CPU, and there was no bus capacitance, then it would generate a perfect rectangular wave SCL signal with fast rise and fall times as shown.

The CPU will start timing the SCL ‘HIGH’ period as soon as it sets the SCL to become ‘HIGH’. The rise time is fast.

The CPU will start timing the SCL ‘LOW’ period as soon as it sets the SCL to ‘LOW’. The fall time is fast.

CPUs should have the capability to generate different timings for the SCL ‘HIGH’ and ‘LOW’ time periods.

The Fast Mode and the Fast Mode Plus specifications both require the Master CPU to make DIFFERENT timings for the High and Low SCL periods to meet the maximum specified bus speed – that is to reach the maximum 400kHz or 1MHz.

If the Master CPU does not have the capability to generate different High and Low periods for SCL, and it can only generate a square wave with equal High and Low periods, then it must use the specified minimum LOW period for both the High and Low periods of SCL.

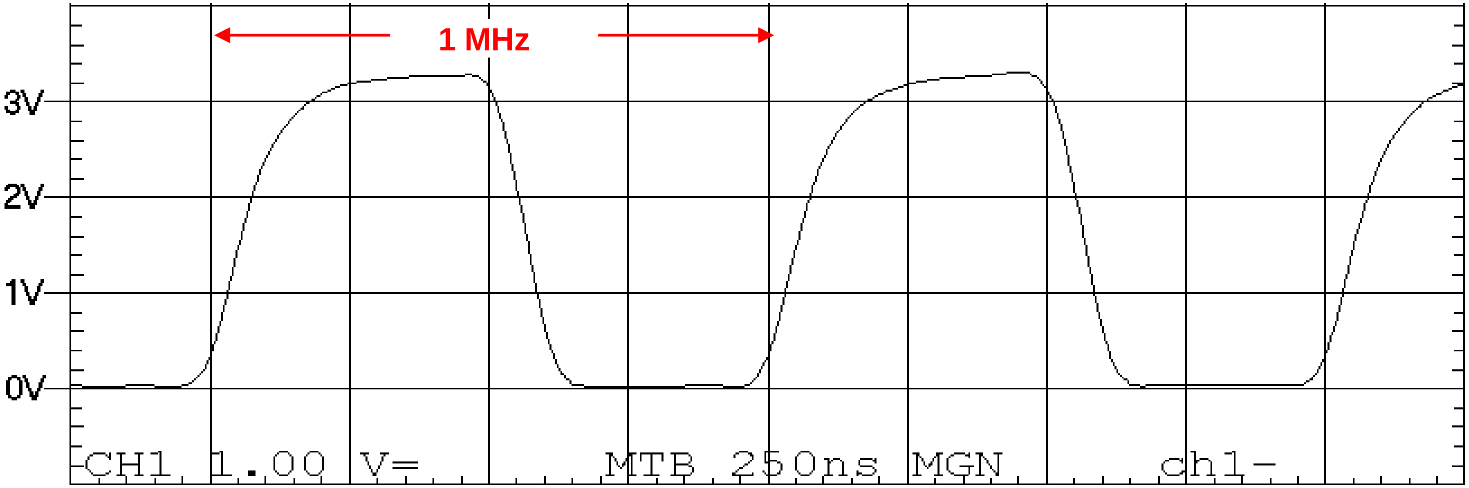
Because the I2C specification for the Low period is longer than the High period, a Master that generates a square wave SCL signal cannot ever run at the maximum I2C speed (400kHz or 1MHz).

Example: Fm+ requires Hi = 260ns, Lo = 500ns, Rise (max) = 120ns, Fall (max) = 120ns. Total = 1000ns. That is 1MHz.

If CPU only generates equal High/Lo periods then High = Lo = 500ns. Rise/Fall = 120ns. Total = 1240ns. That is 806.5kHz

Point 1: The programming capability of the Master CPU will determine the bus speed that can be reached.

Master bus components determine rise and fall times



Example of 1MHz SCL waveform on bus 'A' with the PCA9600 buffer disabled (Vcc=0V)

This oscilloscope trace capture shows the SCL waveform produced by an LPC932 microprocessor that is only specified for operation in Fast Mode (400kHz max) but here it is clocked from an external 16MHz source and programmed to generate an SCL clock at 1MHz having equal High and Low periods.

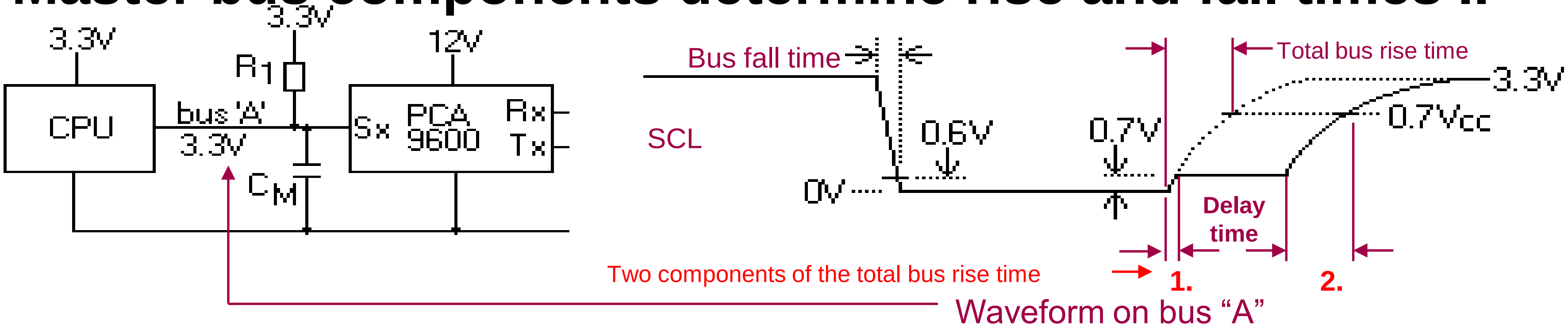
The rise time here is determined by a 1.4k pull-up resistor to 3.3V (2mA max) and is approximately 60ns (I^{2}C).

That is equivalent to an RC product of 70ns so with 1.4k pull-up resistor the capacitance on Bus 'A', including the 12pF scope probe, is about 50pF.
For this trace the PCA9600 supply was set to 0V so the capacitance loading of Sx is included, but the PCA9600 is not active so it does not stretch the clock.
The pull-up resistor is calculated using 2mA because at –40C the Sx pin can add max 1mA to the current the CPU must sink. The total CPU current would then be 3mA.

The fall time here is determined by slew rate limiting of the Fast Mode micro and is about 40ns (I^{2}C).

(Compare with Slide 13 which shows the traces taken with the buffer active. The bus frequency drops to 727kHz due to buffer action and cable rise time delays.)

Master bus components determine rise and fall times II



Bus fall time

Total bus rise time

SCL

Delay time

Two components of the total bus rise time

1.

2.

Waveform on bus “A”

Fall time of bus ‘A’: This is determined by the drive current capability of the master or by its slew controlled fall.

It is reasonable to calculate for Fm+ using a slew rate of approximately 0.2V/ns. When the CPU is driving the Sx input the bus must fall to 0.6V to cause the PCA9600 to switch Tx to low, so the fall time = (Vcc – 0.6V) / 0.2 (ns).
For Vcc = 3.3V, the fall time when the CPU drives the Sx will be approximately 13.5ns.
When the Sx is driving the CPU pin the bus must fall to 0.3Vcc (worst case design) so the fall time = (Vcc x 0.7) / 0.25 (ns)
When the capacitance is large, say >100pF, the fall time can be checked using the formula for charging a capacitor (C) with a current (I) to a voltage (V): C x V = I x t. The V is either (Vccx0.7) or (Vcc-0.6). “I” is the dynamic sink capability of the CPU or the PCA9600. “t” is the fall time. The dynamic sink capability of PCA9600 at Sx/Sy is typ.15mA

Point 2: The **fall** time of this bus will usually NOT have any significant effect on the system timing. It is a small factor.

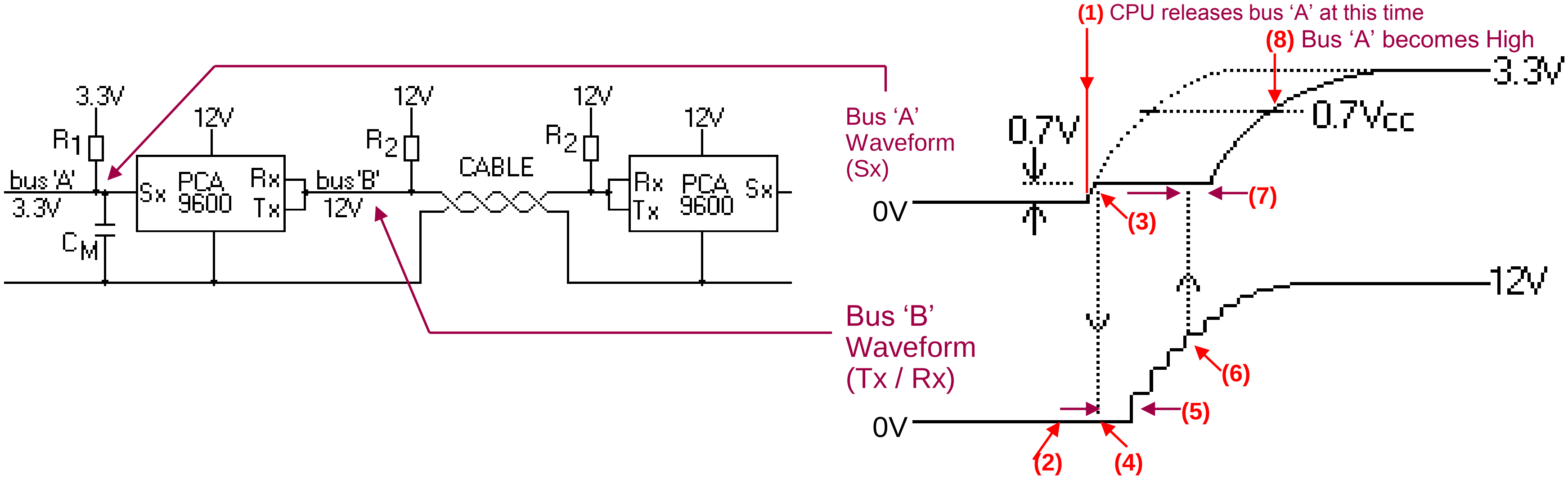
Rise time of bus ‘A’: The total bus ‘A’ rise time is marked on the dot curve showing the normal exponential rise of the bus.
The rise time is determined by the pull-up resistor (R_1) and the total capacitance on this Master bus (C_M). The total rise time is the time to rise from 0V to 0.7Vcc (worst case switching level of the CPU). The RC product determines the conventional rise time, that is, the time for an exponential to rise from 0V to 0.63Vcc.
The total time for the exponential component of the rise from 0V to 0.7Vcc = 1.2 RC. The rise time of bus ‘A’ = $1.2xR_1xC_M$.

The actual rise time has two components **1.** and **2.** The first component is the time to rise from 0V to the switching level of Sx (that is 0.7V). The second component is the time to rise from the 0.7V level to the worst case High level of the CPU, that is 0.7Vcc. Between those two components there is a **Delay Time** as marked.

Delay time of bus ‘A’: The ‘step’ in the rising edge of bus ‘A’ is a **delay** caused by the buffer.
It is not part of the bus ‘A’ rise time, and it is not affected by the components on bus ‘A’.

Point 3: The **total** delay on the rising edge of bus ‘A’ has two parts, a rise time and a delay. This delay is called clock stretching. See Slides 12 and 13 for more detail about clock stretching.

The PCA9600 and the components on the cable bus delay the rising edge of bus ‘A’ after its release by the CPU



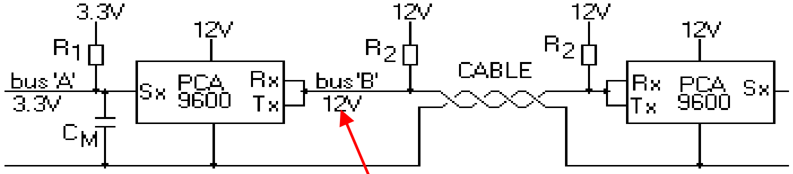
(1) CPU releases bus ‘A’ at this time

(8) Bus ‘A’ becomes High

At the end of the Low period **(1)** the Sx pin is low and that forces Tx and bus ‘B’ to be low **(2).**
Tx is connected to Rx therefore Rx will be low. While Rx is Low, Sx will not allow bus ‘A’ to rise above 0.7V.
When the CPU releases the bus ‘A’ the bus voltage will rise with this sequence:
Bus ‘A’ rises to the switching level **(3)** of Sx (0.6V). When it reaches 0.6V that is the signal for PCA9600 to release Tx **(4)**.
Bus ‘A” continues to rise to 0.7V and is clamped there until Rx is High **(6)**. 0.7V is a ‘High’ input for Sx, but there is an internal propagation delay time **(5)** before Tx actually releases bus ‘B’.
After Tx releases bus ‘B’ that bus takes some time to rise to the logic high switching level **(6)** at Rx.
The time for bus ‘B’ to rise is determined by the two pull-up resistors (R_{2}) and the cable type and length.
After Rx goes High there is an internal propagation delay **(7)** before PCA9600 releases the bus ‘A’ and it can continue its exponential rise from 0.7V to the logic High switching level of the CPU **(8)** (worst case 0.7Vcc).

Point 4: The time delay between the CPU releasing its bus ‘A’ and its bus actually becoming High is called clock stretch.
Because the ‘Low’ period of the clock is increased (stretched), the whole cycle period of the bus clock is increased so **the frequency of the bus clock will be lower than the frequency that the CPU was programmed to deliver.**

The PCA9600 and the components on the cable bus delay the rising edge of bus ‘A’ after its release by the CPU II

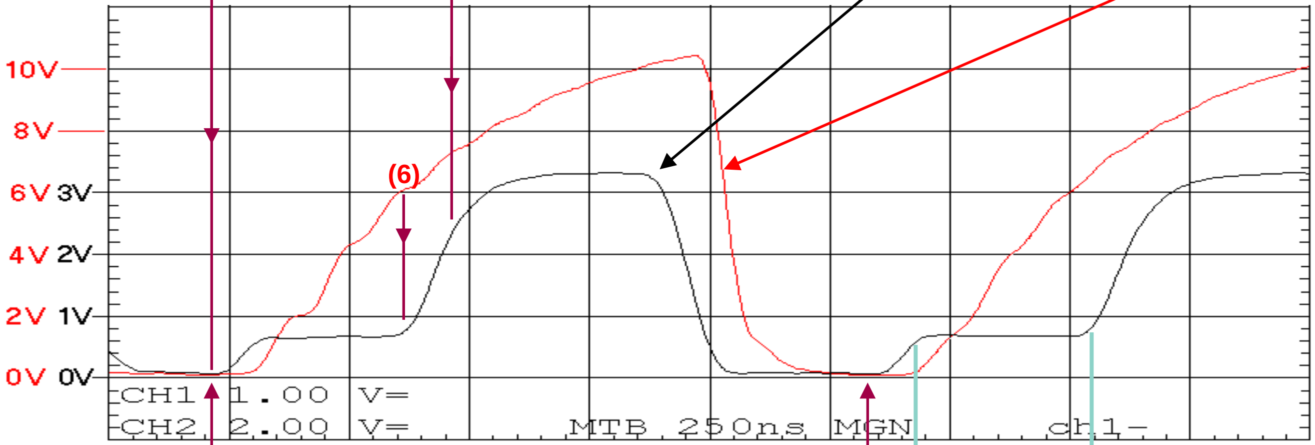


Waveform on bus ‘A”

Waveform on bus ‘B”, at the Master end of cable

(1) CPU releases bus ‘A’ at this time

(8) Bus ‘A’ becomes High



Clock period, with PCA9600 active, = 1375ns, **Freq = 727MHz**

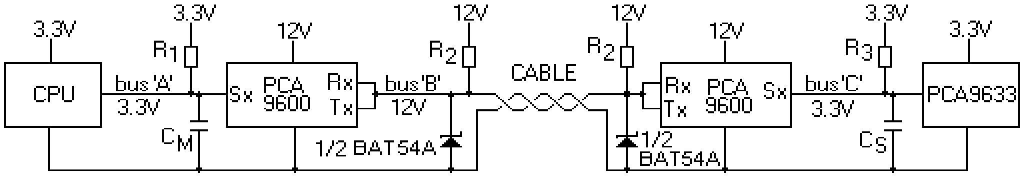
SCL rise delay on bus ‘A’ at the Master caused by buffer and cable.

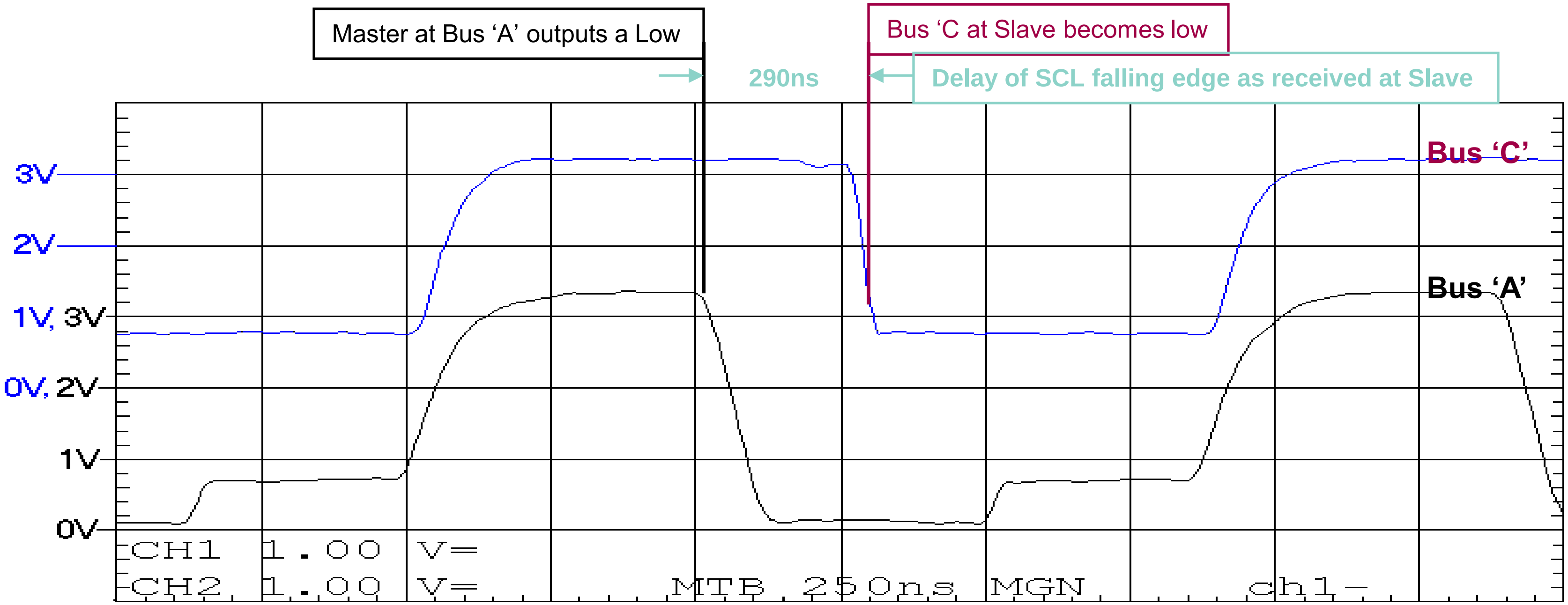
SCL delay added by buffer:

This ‘scope capture shows the waveforms for a **10m Cat5e cable** on bus ‘B’ with R1=1.42k C_M=50pF R2 = 1150 ohms each, resultant = 575 ohms.
Propagation delay in PCA9600 from Sx reaching 0.6V to Tx release = 55ns
Cable capacitance is approx 50pF/m, total approx 500pF. Add 50pF for IC pins, PCB traces, connectors. Total approx 550pF. RC product for cable bus = 316ns.
Cable rise time 0V to 6V = 0.69RC = 218ns. Cable propagation delay at approx 5ns/m = 50ns. Propagation delay from Rx reaching 6V to Sx release =65ns.
Total calculated delay of rising edge of cable signal at slave end of cable approx 55+218+50+65= 388ns = 1.55 graticule divisions.
Scope trace for bus ‘B’ is showing about 1.5 graticule divisions for this delay, shown between the green marking lines on the bus ‘A’ trace.

Point 5: The rise time of the cable bus can be the largest contribution to clock stretching caused by buffers

The SCL low generated by the Master arrives at the slave after delays caused by the cable and buffer




Master at Bus ‘A’ outputs a Low

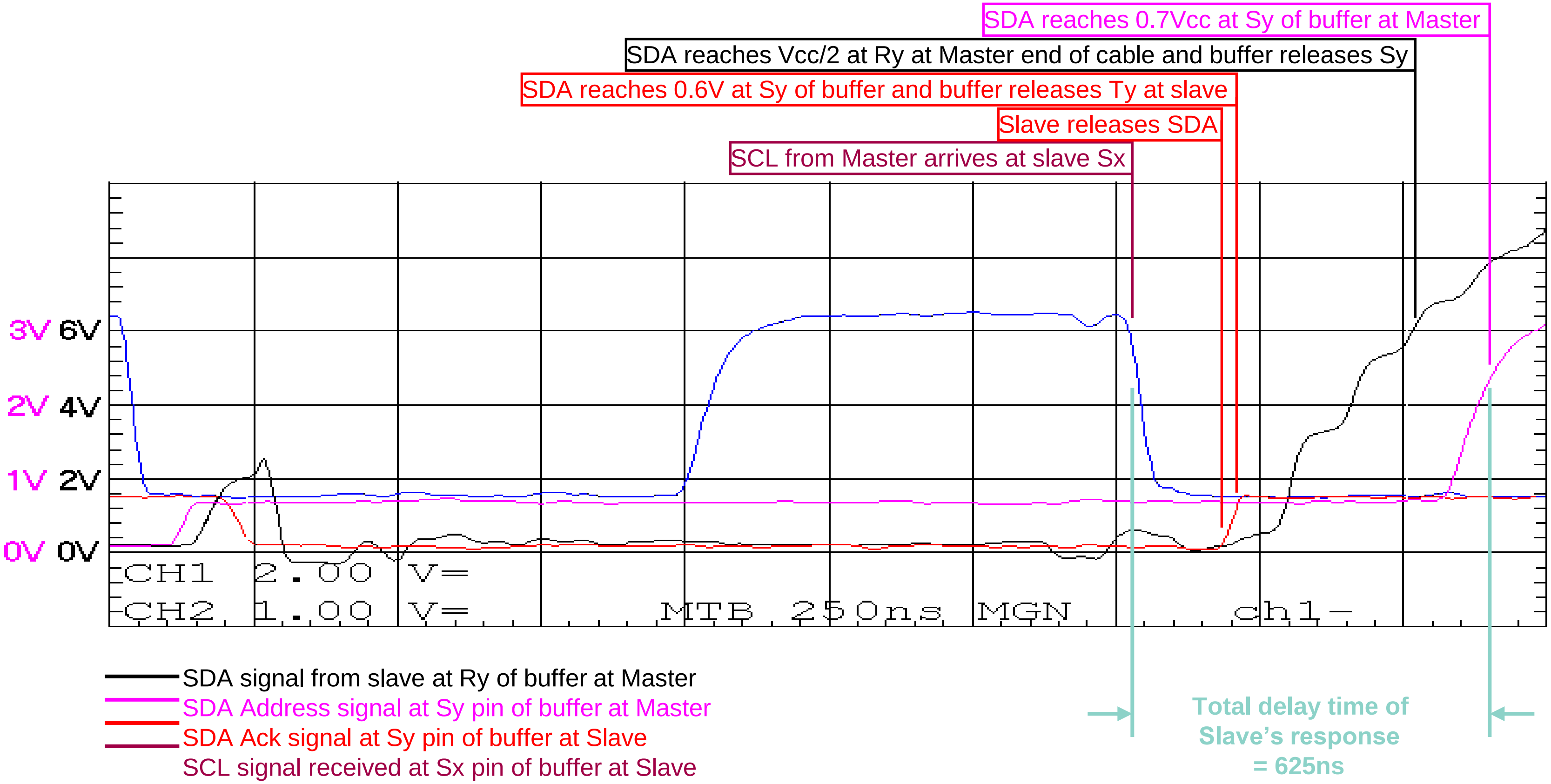
290ns

Bus ‘C at Slave becomes low

Delay of SCL falling edge as received at Slave

Delay of SCL includes the following: Time for bus ‘A” to fall to 0.5V so Sx becomes Low, PCA9600 internal propagation delay before Tx starts to fall, Time for Rx at far end of cable to become low (=PCA9600 Vcc/2), PCA9600 internal propagation delay time after Rx is low until Sx starts to fall, Time for bus ‘C’ to fall to logic low level of the slave (PCA9633, 0.3Vcc worst case). The time for the cable to become low at Rx includes the time for the signal to fall to Vcc/2, and that may depend on signal reflections on the cable.

The slave generated SDA signal returns to the Master



SDA reaches 0.7Vcc at Sy of buffer at Master

SDA reaches Vcc/2 at Ry at Master end of cable and buffer releases Sy

SDA reaches 0.6V at Sy of buffer and buffer releases Ty at slave

Slave releases SDA

SCL from Master arrives at slave Sx

SDA signal from slave at Ry of buffer at Master

SDA Address signal at Sy pin of buffer at Master

SDA Ack signal at Sy pin of buffer at Slave

SCL signal received at Sx pin of buffer at Slave

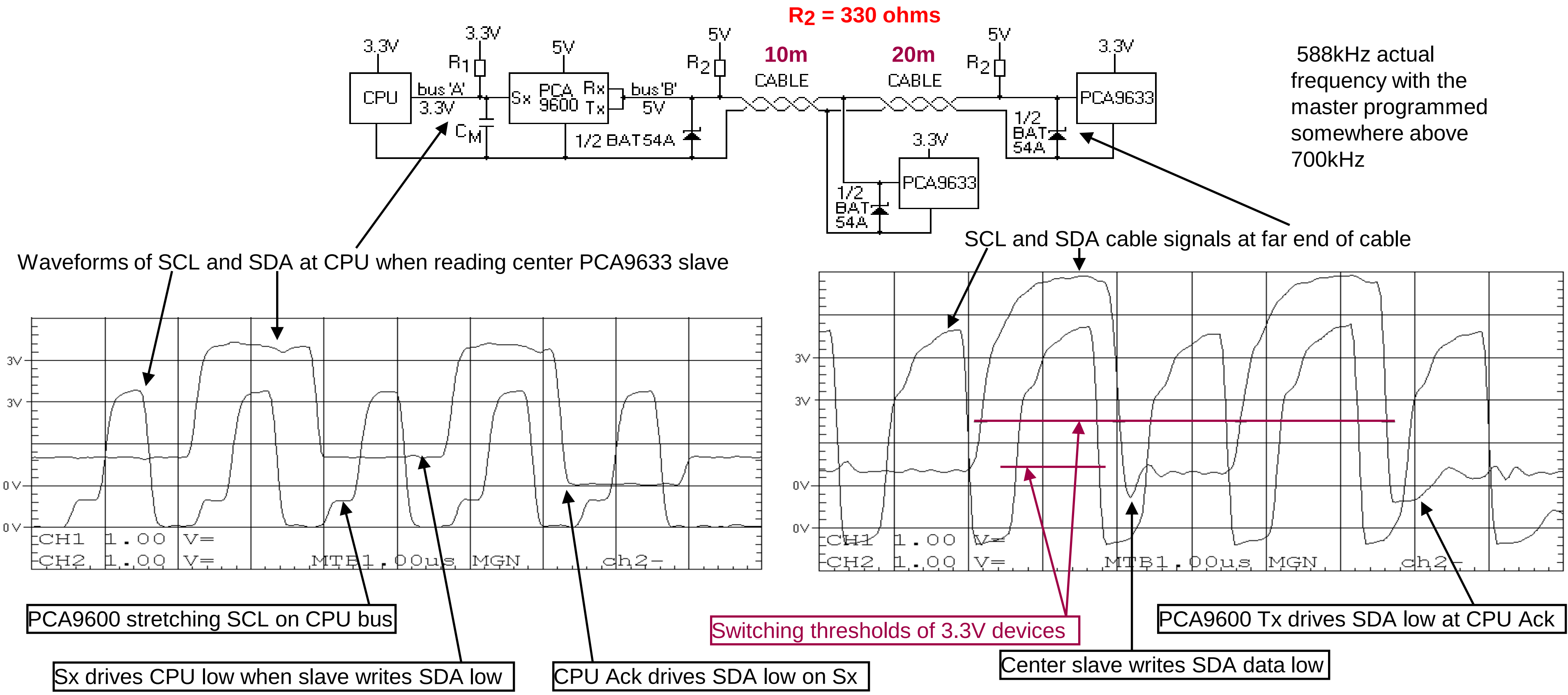
Total delay time of Slave's response = 625ns

The total delay time for the slave data response to be valid at the Master includes the following:
Time for the slave to respond with data after the SCL goes low (see red trace delays relative to blue SCL falling edges)
Time for slave data at Sx of PCA9600 to propagate to Tx pin (see black trace delays after red trace changes from 0V to 0.6V or 0.6V to 0V)
Time for high level at Rx (Vcc/2) to propagate to Sx (delay between black trace reaching 6V and magenta trace starting to rise)
Time for Sx at Master to rise from 0V to High level (worst case 0.7Vcc). (Rise time of magenta trace)

The next rising edge of the SCL at the Master (n.b. NOT shown in this set of traces) must not occur until after the SDA is guaranteed valid for the required set-up time.

(This SDA signal delay when the slave outputs a Low will generally be less than for a High due to the longer rise time on the cable bus so worst case designs are usually calculated using the rising data signal delay as shown on the right hand side of these traces)

Multi-drop arrangement with direct drive to Fm+ slaves



588kHz actual frequency with the master programmed somewhere above 700kHz

Waveforms of SCL and SDA at CPU when reading center PCA9633 slave

SCL and SDA cable signals at far end of cable

PCA9600 stretching SCL on CPU bus

Sx drives CPU low when slave writes SDA low

CPU Ack drives SDA low on Sx

Switching thresholds of 3.3V devices

PCA9600 Tx drives SDA low at CPU Ack

Center slave writes SDA data low

Suggested arrangement for **multi-drop buses**, including directly driving 5V tolerant Fm+ slaves:

1) Always fit BAT54A Schottky diode clamps on SCL and SDA signals at **every** node.

2) Use strongest allowed pull-up on cable, divided equally between start and extreme end of cable.
(so each pull-up R_2 (minimum) = 2 x (cable supply – 0.4V / 30mA). For Vcc= 5V Rmin = 307 ohms. Example uses 330 ohms.)

3) PCA9600 or P82B96 (with linked Tx/Rx) may also be included on cable at any node.
Connect each buffer's Vcc to the cable bus voltage (and 5V is preferred even if Fm+ slaves are not directly attached)

FAQ

1. What about doing this for multi-drop bus systems?

   Answer > That's really tough because it has so many variables and we doubt we can attempt it, but we will provide in the future some guidelines about what is likely to work.

2. What about direct interfacing to Fm+ slaves with the P82B96 or PCA900 (i.e., using a buffer at only ONE end of the cable to the normal strength devices and having Fm+ devices (e.g., PCA96XX) on the other ends of the long cables.

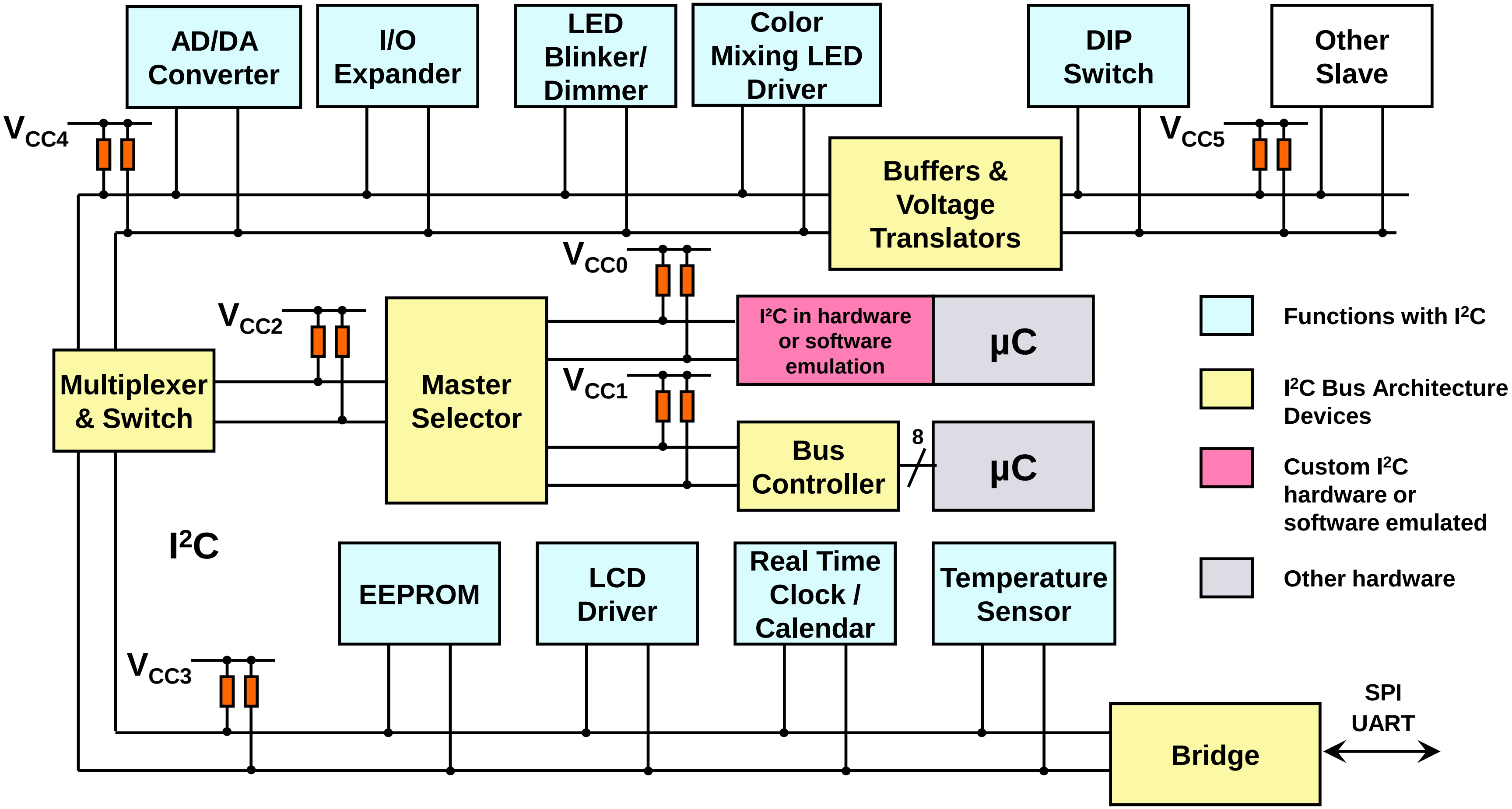
   Answer > This is one of the intended applications of the Fm+ devices and we’ll need to provide a better answer about the cable drive capability of Fm+ slaves later.

3. Will anyone be investigating whether Fm+ slaves can interface directly with long cables (e.g., PCA9665 driving directly long point to point or multiple point cables to PCA96XX slaves)?

   Answer > Possibly you noticed the LAST slide in the set shows how the long cables work, was a brief attempt to address multi-drop Fm+ slaves direct on a long cable. It sets some guidelines. It worked well so we included an example of the nice waveforms obtained with a 30m cable running 5V signals. It was driven by PCA9600 but every indication was that PCA9665 should also be fine since the output drive is the same as PCA9633. Don't know how much longer than 30m people might expect to run but that clearly isn't the limit, it's still looking good. The unknowns for a REAL CMOS system (not 30m safely coiled up in the lab) remain things like ESD and other induced transient (nearby lightning) tolerance, and things like ground potential differences. I'm surprised, given the very large and fast growing number of examples around the environment, that there have not been questions about using the Fm+ drivers for architectural lighting.

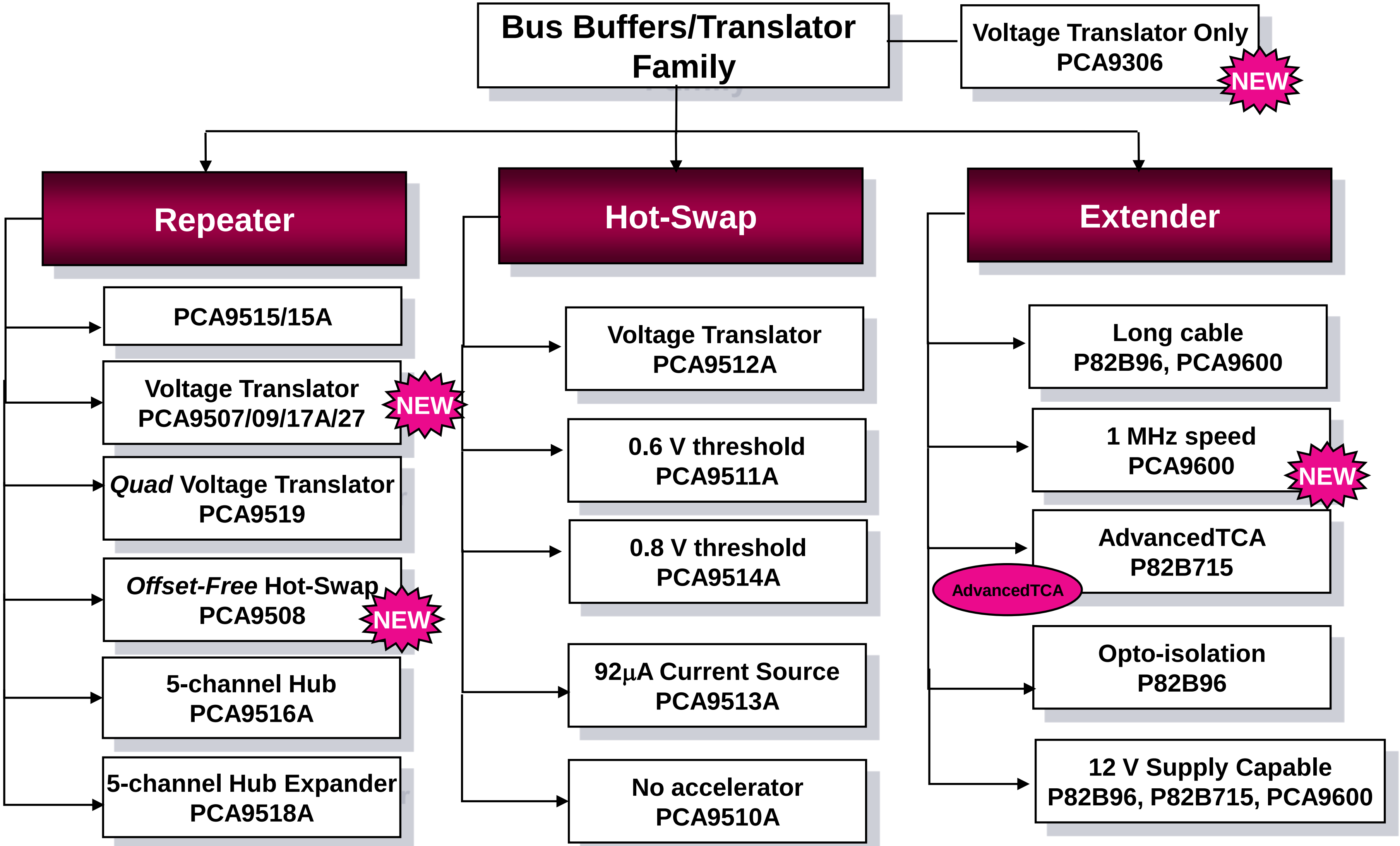
NXP I^{2}C Device overview with focus on long distance Bus Buffer

I²C Portfolio



🟦	Functions with I²C
🟨	I²C Bus Architecture Devices
🟥	Custom I²C hardware or software emulated
⬜	Other hardware

Buffers/Voltage Translators Family



Bus Buffers/Translator Family

Voltage Translator Only
PCA9306
NEW

Repeater

- PCA9515/15A
- Voltage Translator
PCA9507/09/17A/27
NEW
- *Quad* Voltage Translator
PCA9519
- *Offset-Free* Hot-Swap
PCA9508
NEW
- 5-channel Hub
PCA9516A
- 5-channel Hub Expander
PCA9518A

Hot-Swap

- Voltage Translator
PCA9512A
- 0.6 V threshold
PCA9511A
- 0.8 V threshold
PCA9514A
- 92µA Current Source
PCA9513A
- No accelerator
PCA9510A

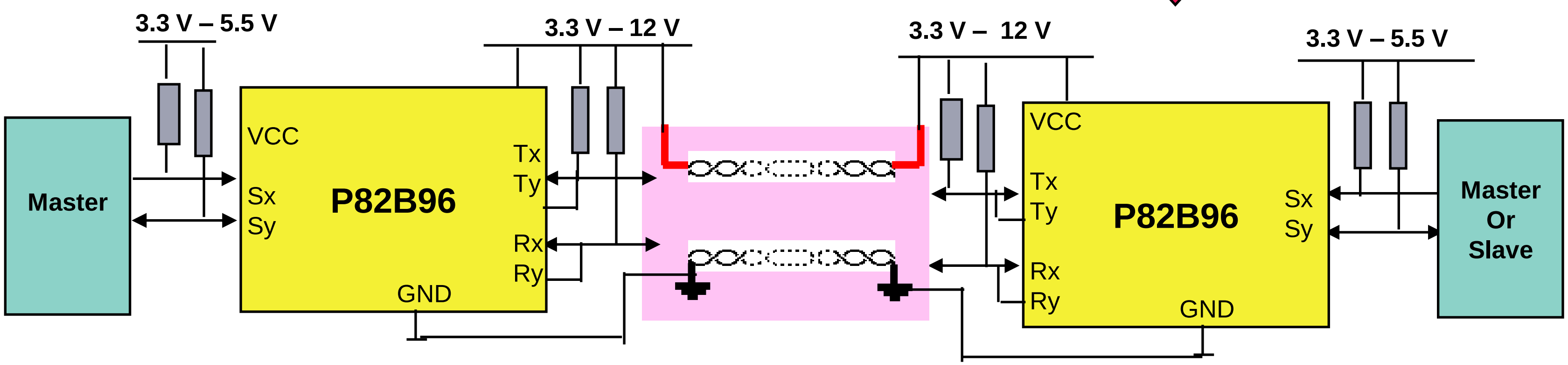
Extender

- Long cable
P82B96, PCA9600
- 1 MHz speed
PCA9600
NEW
- AdvancedTCA
P82B715
AdvancedTCA
- Opto-isolation
P82B96
- 12 V Supply Capable
P82B96, P82B715, PCA9600

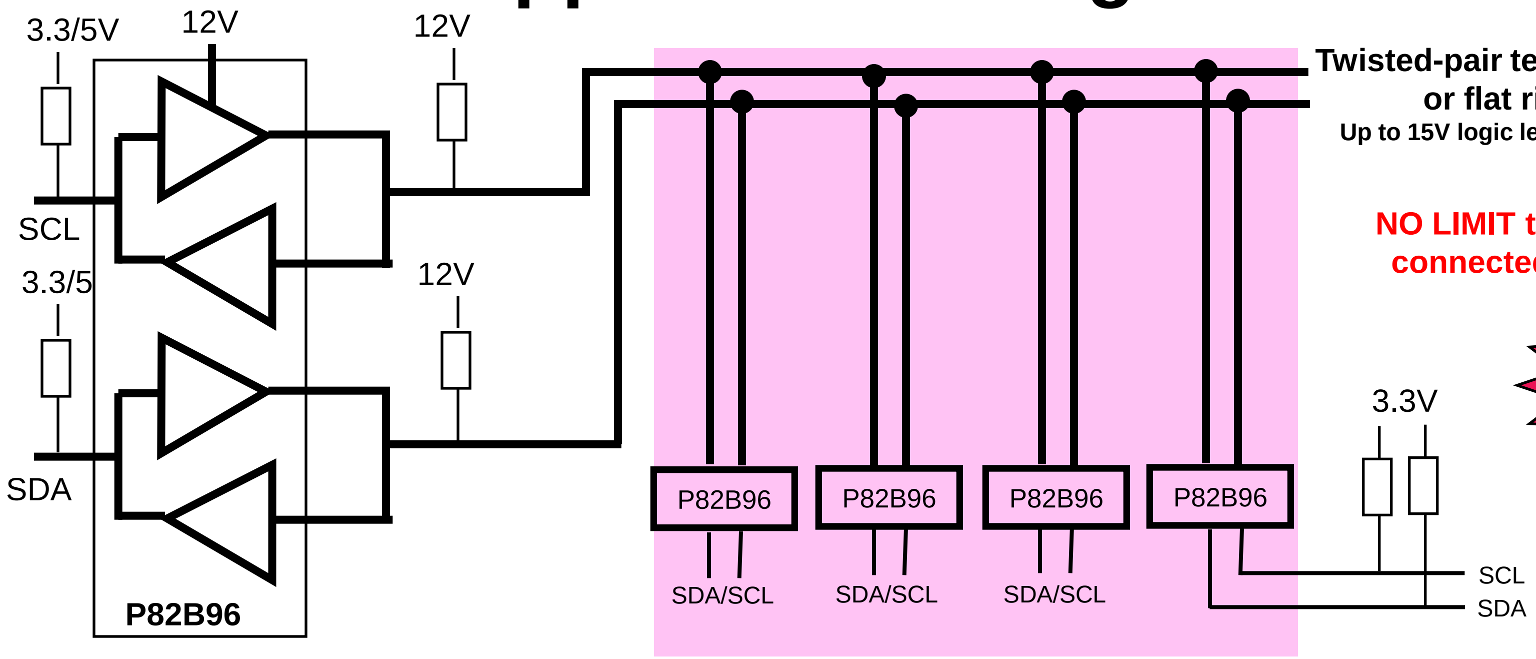
I²C over Long Cable (> 1 kilo meter)

- **Problem**: Driving > 1 km typically requires very expensive discrete components
- **Solution**: I²C-bus is a cost effective solution and allows bidirectional communication up to 60 kHz – and with P82B96 or newer higher speed drop in replacement PCA9600 bus extender, it has high drive on Tx/Ty, Rx/Ry side, is inexpensive and reliable.

New PCA9600



Industrial Application Using I²C bus Extender

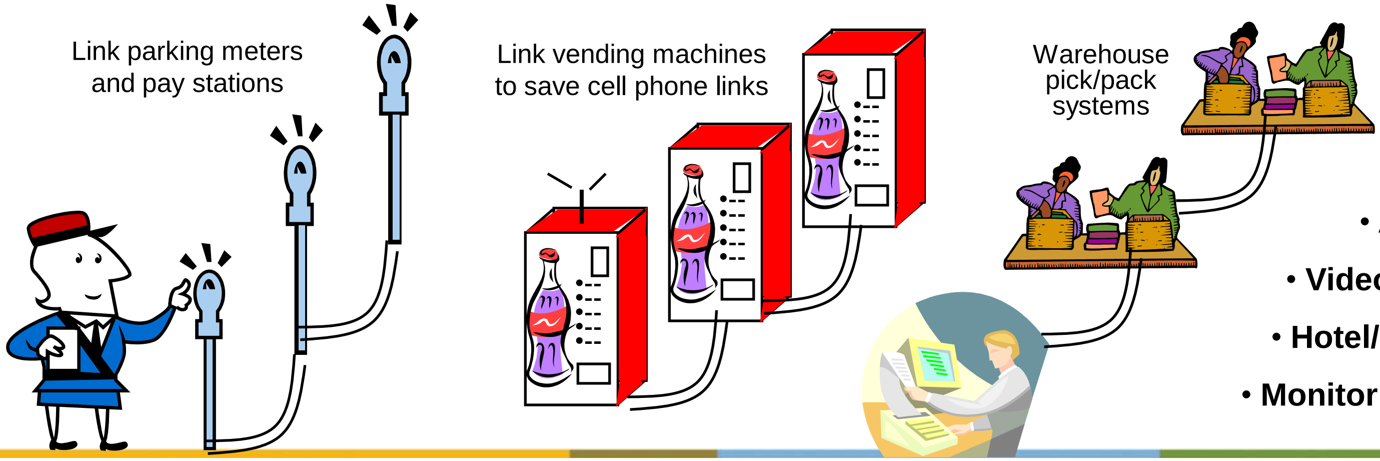


Twisted-pair telephone wires, USB or flat ribbon cables

Up to 15V logic levels, Include V_{CC} & GND

NO LIMIT to the number of connected bus devices !

New PCA9600



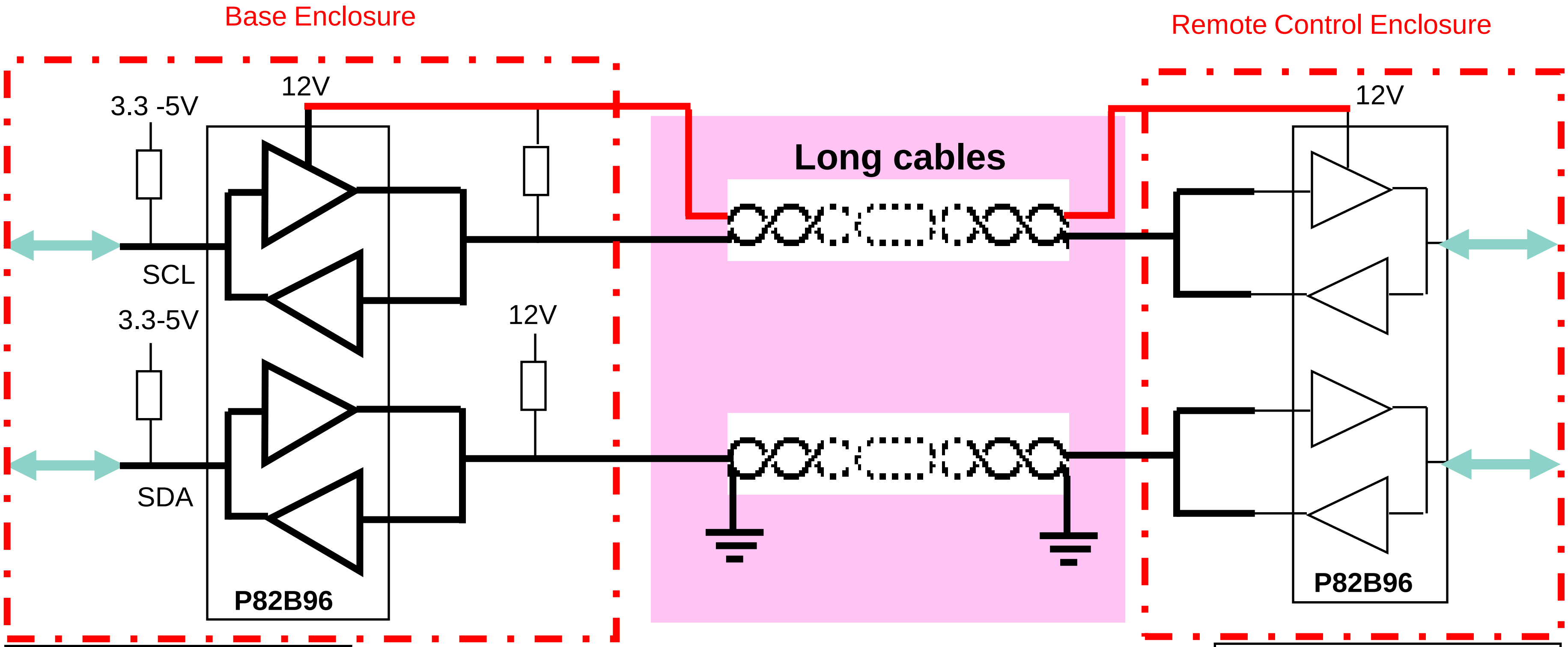
Link parking meters and pay stations

Link vending machines to save cell phone links

Warehouse pick/pack systems

- **Factory automation**
- **Access/alarm systems**
- **Video, LCD & LED display signs**
- **Hotel/motel management systems**
- **Monitor emergency lighting/exit signs**

Driving I^2C Bus Signals Long Distances



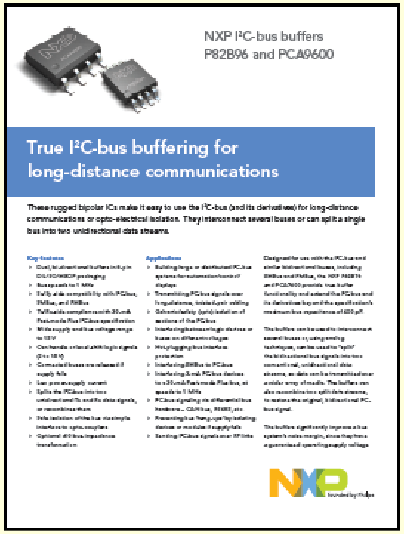
- Normal I^2C logic levels (3.3 or 5 V)
- I^2C currents (3mA)

New PCA9600 allows up to 1 MHz

- Conventional CMOS logic levels (2-15V)
- Higher current option, up to 30mA static sink

- Normal I^2C logic levels (3.3 or 5 V)
- I^2C currents (3mA)

Literature – P82B96 & PCA9600



NXP I^{2}C-bus buffers
P82B96 and PCA9600

True I^{2}C-bus buffering for long-distance communications

These rugged bipolar ICs make it easy to use the I^{2}C-bus (and its derivatives) for long-distance communications or opto-electrical isolation. They interconnect several buses or can split a single bus into two unidirectional data streams.

Key features

[illegible]

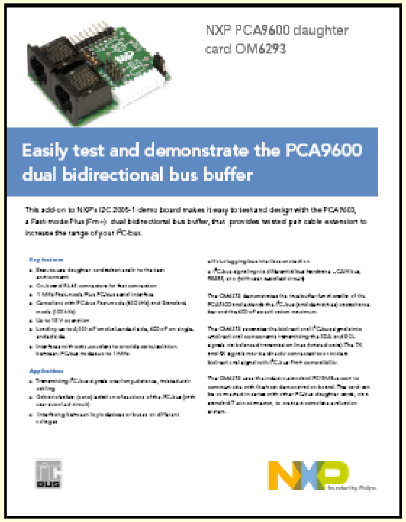
Applications

[illegible]

[illegible]

P82B96 & PCA9600 Product Information

www.standardics.nxp.com/literature/other/i2c/pdf/bus.buffers.p82b96.pca9600.pdf



NXP PCA9600 daughter card OM6293

Easily test and demonstrate the PCA9600 dual bidirectional bus buffer

This add-on to NXP's I2C 2005-1 demo board makes it easy to test and design with the PCA9600, a Fast-mode Plus (Fm+) dual bidirectional bus buffer, that provides twisted pair cable extension to increase the range of your I^{2}C-bus.

Key features

[illegible]

Applications

[illegible]

[illegible]

PCA9600 Daughter Card

www.standardics.nxp.com/literature/leaflets/i2c/pdf/pca9600.daughter.card.pdf

Literature – P82B715

NXP I²C-bus extender
P82B715

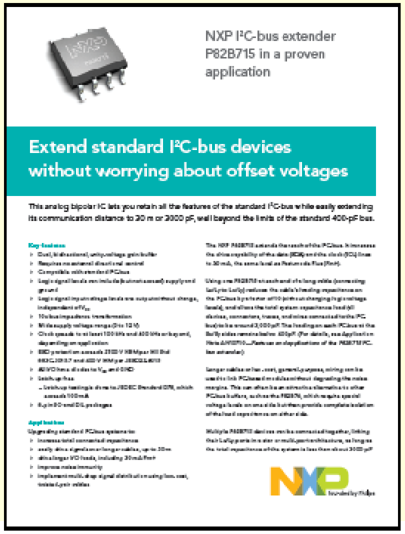
Extend the reach of any I²C-bus system without special offset voltage levels

This analog bipolar IC retains all the features of the I²C-bus including interoperability with all masters/slaves and bus buffers while extending the I²C-bus well beyond the limits of the standard 400-pF bus capacitance without having to use special offset voltage levels and their inherent loss in noise margin.

[illegible]

P82B715
Product
Information

www.standardics.nxp.com/literature/leaflets/i2c/pdf/p82b715.pdf



NXP I²C-bus extender
P82B715 in a proven application

Extend standard I²C-bus devices without worrying about offset voltages

This analog bipolar IC lets you retain all the features of the standard I²C-bus while easily extending its communication distance to 30 m or 3000 pF, well beyond the limits of the standard 400-pF bus.

[illegible]

P82B715
No offset -
Technical
Information

www.standardics.nxp.com/literature/leaflets/i2c/pdf/p82b715.application.pdf

Contact

- **General**: www.nxp.com/support

- **I^{2}C**:

  www.standardics.nxp.com/i2c

  I2C.Support @ nxp.com (E-mail Support)

  forums.nxp.com/forums/viewforum.php?f=6 (Forum)

  www.standardics.nxp.com/support/documents/i2c/ (Application Notes)

- **IBIS and HSPICE Model**

  www.standardics.nxp.com/support/models