Designing with I²C-Bus Devices

Emmanuel T. Nana
Technical Marketing Manager
Secure Interfaces & Power

June 2019 | Session #AMF-IND-T3704
# NXP Secure Interfaces & Power Solutions

## Signal Integrity & Routing Solutions
- **Signal Switches & Re-drivers**
  - USB 3.1, USB Type-C
  - Thunderbolt
  - PCIe, SATA, SAS
  - DP, HDMI, VGA
  - Audio, Data
  - Memory Interface

  *Industry leader in high-speed switching. Lowest-power consumption re-drivers*

## Load Switches
- **Over Voltage Protection**
- **Over Current Protection**
- **Reverse Current Protection**
- **Under voltage Lockout**
- **Thermal Shutdown**
- **Low $R_{ON}$**
- **Low Quiescent Current**

  *HV Load switching with 100V surge protection.*

## Interface Solutions
- **DisplayPort Bridges**
- **UARTS**
- **Comparators**
- **I²C Bus Buffers**
- **I²C Bus Controllers**
- **I²C Mixes & Switches**
- **Voltage Level Translators**

  *Industry's largest I²C Portfolio for Mobile, Computing and Industrial.*

## Wireless Connectivity & Smart Sensor Solutions
- **NTAG Smart Sensors**
- **NFMI Radio**
- **Audio over BLE**
- **RF & IF Discretes**
  - Transceivers
  - LNAs
  - Mixers
  - Switches

  *Integrated temperature logging solutions. Ultra low-power single-chip solution, providing robust wireless audio streaming.*

## Security & Authentication
- **Anti-Counterfeit Solution**

  *Industry’s smallest package with lowest power.*

## Power Solutions
- **USB Power Delivery**
- **AC-DC Controllers**
- **DC-DC Boost Converters**
- **Direct Charging (Rapid Battery Charging)**
- **Wireless Charging (Qi/AWPP)**
- **Micro-PMIC**
- **Powerline Communication Modern**

  *High efficiency power conversion. Support of multi-charging protocols (Direct, USB-PD, QC, BC1.2, and proprietary).*

## Bus Peripherals
- **Real Time Clocks**
- **GPIO Expanders**
- **Temperature Sensors**
- **LCD Drivers**
- **LED Controllers**
- **Capacitive Sensors**
- **Stepper Motor Controllers**
- **EEPROM**
- **Watch IC**
- **Data Converter**
- **DIP Switches**

  *Ultra low-power RTC’s. Widest portfolio of GPIO Expanders.*

## Smart Audio Solutions
- **Class AB Amplifiers**
- **Class D Amplifiers**
- **Smart Amplifiers (w integrated DSP)**
- **Software**
- **Speaker Protection**
- **Audio DAC & ADC**

  *Best-in-class speaker protection hardware and software Class D Amplifier solutions.*
Agenda

- Introduction to I²C
- I²C-Bus Communication Protocol
- I²C-Bus Pull-up Resistor Calculation
- I²C Interface Signals
- I²C-Bus Tools
Introduction to I²C-Bus
What is I²C?

• A communication bus for slow speed digital data
• I²C = Inter Integrated Circuit (Philips invented in the 1980s)
• Original purpose to link a CPU to other circuits in television sets
• Links one or more SLAVE devices
• To a MASTER (one or more BUS CONTROLLERS)
What is I²C?

- I²C BUS can have:
  - Multiple masters
  - Multiple slaves
- Only one master talks to one slave at a time
- All the slaves on the same bus must have different address
- Slow speed device cannot understand higher speed transfer
I\textsuperscript{2}C-Bus Physical Layer

Physical Layer = Electrical Connections

Two Wires: Data and Clock (*plus ground and supply*)

- Data line (SDA)
- Clock line (SCL)

Pull Up Resistors on each line
I²C-Bus Communication Protocol
What is the I²C Protocol Layer?

Protocol Layer = Data Format, Traffic, Collision Arbitration

An I²C Bus must have:
- Two node types (Master and Slave)
- Minimum of ONE Slave and ONE Bus Master
**I²C Interface Protocol**

I²C BUS constructs off 9 bit block

START condition:
When SCL is HIGH then SDA goes from HIGH to LOW

Address bit
7 bit after START condition

Read or Write bit
After 7 bit address, the 8th bit is Read or Write bit
1 = Read cycle or 0 = Write cycle

ACK
Synchronization bit between master and slave
0 = ACK and 1 = NACK

Data Byte
8-bit after address byte is data byte from master or slave

STOP condition:
When clock line (SCL) is HIGH then the data line (SDA) goes LOW to HIGH

---

![I²C Interface Protocol Diagram](image-url)
Complete I²C Interface Protocol

I²C protocol

Start with START condition

Bit 8\textsuperscript{th} is Write/Read command

Bit 9\textsuperscript{th} is a synchronization bit

7-bits slave address

8-bits of data

End with STOP condition

Slave

Slave address

Reg 8bit

NACK
I²C-Bus Pull-up Resistor Calculation
I/O (SDA & SCL) Driver Architecture

SDA and SCL are open drain/collector
• Required pull-up resistors to pull the line to logic “1”
Key Electrical Parameters

<table>
<thead>
<tr>
<th>Bit Rate (kb/s)</th>
<th>Standard Mode</th>
<th>Fast Mode</th>
<th>Fast Mode Plus</th>
<th>High Speed Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 100</td>
<td>0 to 400</td>
<td>0 to 1000</td>
<td>0 to 1700</td>
<td>0 to 3400</td>
</tr>
</tbody>
</table>

| Max Load (pF)  | 400           | 400       | 560            | 400             | 100             |
| Rise time (ns) | 1000          | 300       | 120            | 160             | 80              |
| Noise filter (ns) | -          | 50        | 10             | 10              | 10              |

![Diagram showing Key Electrical Parameters](image-url)
Calculating Pull-up Resistors

1) \( R_{\text{MIN}} < R_{\text{PU}} < R_{\text{MAX}} \)

2) \( R_{\text{MIN}} = \frac{(V_{\text{DDMAX}} - V_{\text{OLMAX}})}{I_{\text{OLMAX}}} \)

<table>
<thead>
<tr>
<th>( V_{\text{DDMAX}} )</th>
<th>( V_{\text{OLMAX}} )</th>
<th>( R_{\text{MIN}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.7 V</td>
<td>0.6 V</td>
<td>700 ( \Omega )</td>
</tr>
<tr>
<td>3.6 V</td>
<td>0.6 V</td>
<td>1.0 k( \Omega )</td>
</tr>
</tbody>
</table>

3) \( R_{\text{MAX}} * C_{\text{MAX}} = 1.18 * t_r \)

<table>
<thead>
<tr>
<th>MODE</th>
<th>Frequency</th>
<th>( t_r )</th>
<th>( C_{\text{MAX}} )</th>
<th>( R_{\text{MAX}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>100 kHz</td>
<td>1000 ns</td>
<td>400 pF</td>
<td>2.96 k( \Omega )</td>
</tr>
<tr>
<td>Fast Mode</td>
<td>400 kHz</td>
<td>300 ns</td>
<td>400 pF</td>
<td>885 ( \Omega )</td>
</tr>
<tr>
<td>Fast Mode Plus</td>
<td>1000 kHz</td>
<td>120 ns</td>
<td>560 pF</td>
<td>252 ( \Omega )</td>
</tr>
</tbody>
</table>

Glossary
- \( R_{\text{PU}} \): Pull-up resistor
- \( R_{\text{MIN}} \): Minimum pull-up resistor
- \( R_{\text{MAX}} \): Maximum pull-up resistor
- \( V_{\text{DDMAX}} \): Maximum supply rail
- \( V_{\text{OLMAX}} \): Maximum output voltage low
- \( I_{\text{OLMAX}} \): Maximum sink current
- \( C_{\text{MAX}} \): Maximum load capacitance
- \( t_r \): Rise time
How to Calculate the I²C-Bus Pull-up Resistors?

Minimum value

There is a minimum resistor value determined by the I²C spec limit of 3 mA

\[ R = \frac{V_{dd_{\text{max}}} - V_{ol_{\text{max}}}}{0.003A} \]

Example: using a 5±0.5 V bus: \[ R = \frac{(5.5V - 0.4V)}{0.003A} = 1.7 \text{ kΩ} \]

Maximum value

Determined by the I²C-bus rise time requirements:

\[ V(t1) = 0.3*V_{dd} = V_{dd} (1-1/e^{t1/RC}); \text{ then } t1 = 0.3566749*RC \]

\[ V(t2) = 0.7*V_{dd} = V_{dd} (1-1/e^{t2/RC}); \text{ then } t2 = 1.2039729*RC \]

\[ t = t2-t1 = 0.8472979*RC \]

For standard-mode I²C-bus: \( t \) = rise time = 1000ns (1 µs)

so \( RC = 1180.2 \text{ ns} \)

Example: at a bus load of 400 pF: \[ R_{\text{max}} = 2.95 \text{ kΩ} \]

For fast-mode:

I²C-bus rise time = 300 ns @ 400 pF: \[ R_{\text{max}} = 885 \text{ Ω} \]
How Does User Derive the Rise Time for I²C-Bus?

I²C-bus rise time is determined as in the following:

1) \[ V(t_1) = 0.3V_{DD} = V_{DD}(1 - \frac{1}{et_1/RC}) \rightarrow t_1 = 0.3566749*RC \text{ (EQ1)} \]

2) \[ V(t_2) = 0.7V_{DD} = V_{DD}(1 - \frac{1}{et_2/RC}) \rightarrow t_2 = 1.2039729*RC \text{ (EQ2)} \]

3) Subtract EQ1 from EQ2 \[ t_{\text{rise time}} = t_2 - t_1 = 0.8472979*RC \]
   \[ R*C = 1.18*t_{\text{rise time}} \]
Effect of Pull-up Resistors

• **Minimum pull-up resistor** limits the maximum current sink that affects the voltage output low (VOL).
  - Increasing pull-up resistor above RMIN leads to decreasing VOL and higher noise margin
  - Decreasing pull-up resistor below RMIN leads to increasing VOL and lower noise margin

• **Maximum pull-up resistor** affects the rise time and speed
  - Increasing pull-up resistor above RMAX leads to slower/possible rise time violation or lower speed
  - Decreasing pull-up resistor below RMAX leads to faster rise time and speed
Bus Loading and Timing Relationship

The \( \text{I}^2\text{C} \) bus specifications require certain bus rise times. Those times are defined with the time measured between the bus LOW and HIGH limit levels of 30% and 70% of VDD.

From the expression \( V(t) = V_{DD}(1 - 1/e^{t/\tau}) \), and as shown on the curve at the right, the time for the bus to rise to 30%VDD is \( 0.357\tau \).

The time to rise to 70%VDD is \( 1.204\tau \), so the \( \text{I}^2\text{C} \) rise time, to rise from 30% to 70% of VDD, is \( (1.204 - 0.357) = 0.847\tau \).

Because \( \tau = RC \), meeting the \( \text{I}^2\text{C} \) rise time requirements means the pull-up “R” and bus capacitance “C” must satisfy the relationship:

\[
0.847\ RC \leq \ \text{I}^2\text{C} \text{ rise time specification}
\]

or \( \ RC \leq \ \text{I}^2\text{C} \text{ rise time specification} / 0.847 \)

or \( \ RC \leq \ \text{I}^2\text{C} \text{ rise time specification} \times 1.18 \)

For example, to meet the Fast-mode 300 ns rise requirement, the RC product must be less than \( 1.18 \times 300 = 354 \) ns.
I²C Interface Signals
Comparison of \( \text{I}^{2}\text{C} \) Bus logic levels for different supply voltages
“Three Level” I²C Signals

- The I²C Bus has two logical levels (zero and one)
- There are now 3 signal levels, but only 2 logical levels
- This is caused by different strength Drivers, or by Bus Buffer “Offset”
I²C Signals (Oscilloscope Plot)

Start → ACK → stop

SCL (Clock)

SDA (Data)
I²C-Bus Tools
Fm+ I²C-Bus Demonstration System

PC/GUI
- USB
  - The Boardshop Win-I2CUSB DLL Dongle

MASTER
- USB
  - Total Phase Aardvark
  - OM13260 - Fm+ I²C-BUS DEVELOPMENT BOARD

SLAVE
- Wireless adapter
  - OM13303 - GPIO TARGET
  - OM13399 - BRIDGE BOARD
  - OM13487 – 16-bit GPIO Daughter Card

DIP ADAPTER BOARD
- OM13398 – PCA9617A BUS BUFFER BASE BOARD
- OM13487 – LM75 type TS Daughter Card
# Fm+ I²C-Bus Development Board Kit (OM13320)

## Box Content

<table>
<thead>
<tr>
<th>OM number</th>
<th>12 NC Number</th>
<th>Description (50 Chars max)</th>
<th>Description (35 Chars max)</th>
<th>NXP Device Cross Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>OM13260</td>
<td>9353 9591 44588</td>
<td>Fm+ I²C-Bus Development Board (RoHS)</td>
<td>Fm+ I²C-bus Dev Bld (RoHS)</td>
<td>PCA9663, PCA9672, PCA9601, PCA9655</td>
</tr>
<tr>
<td>OM13333</td>
<td>9352 9591 15568</td>
<td>GPIO Target Board (RoHS)</td>
<td>GPIO Target Bld (RoHS)</td>
<td>none</td>
</tr>
<tr>
<td>OM13306</td>
<td>9353 9207 44588</td>
<td>PCA9617A Bus Buffer Demo Bld (RoHS)</td>
<td>PCA9617A Bus Buffer Bld (RoH)</td>
<td></td>
</tr>
<tr>
<td>OM13399</td>
<td>9353 9207 15568</td>
<td>Bridge Board (Spin to Fm+) (RoHS)</td>
<td>Bridge Board (Spin to Fm+) (RoH)</td>
<td></td>
</tr>
</tbody>
</table>

### Fm+ Development Board Kit

- Explore the I²C Bus
- Run demonstrations of NXP’s I²C Fm+ Slaves and Bus Controllers
- Develop I²C Hardware
- Expand this kit with add-on I²C Daughter Cards

**BOX CONTENTS:**

**OM13260 Fm+ Development Board**
- OM13303 GPIO Target BRD (x2)
- OM13398 PCA9617A Bus Buffer Demo Board
- OM13399 Bridge Board
- Cables and Mounting Hardware
Fm+ I²C-Bus Development Board Kit (OM13320)

GPIO Target Board OM13303 (2x)

Bus Buffer Board OM13398

Fm+ Development Board OM13260

Bridge Board OM13399
DIP Adapter Boards

Breakout Board (A through G)
- Prepare any I²C device as a DIP module for the Fm+ I²C Bus EVM Board OM13490

<table>
<thead>
<tr>
<th>OM number</th>
<th>12 NC Number</th>
<th>Description (50 Chrs max)</th>
<th>Description (35 Chrs max)</th>
<th>NXP Device Cross Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>OM13491</td>
<td>Breakout Board A (RoHS)</td>
<td>Breakout Board A (RoHS)</td>
<td>SD8, VSOP8, XQFN8, HVSON8, MSOP8</td>
<td></td>
</tr>
<tr>
<td>OM13492</td>
<td>Breakout Board B (RoHS)</td>
<td>Breakout Board B (RoHS)</td>
<td>TSOP6, WLCS6P6, HVSON6, TSSOP8, ( \times )SON8U, HVSON10, MSOP10</td>
<td></td>
</tr>
<tr>
<td>OM13493</td>
<td>Breakout Board C (RoHS)</td>
<td>Breakout Board C (RoHS)</td>
<td>DQFN14, 16, 20 and 24</td>
<td></td>
</tr>
<tr>
<td>OM13494</td>
<td>Breakout Board D (RoHS)</td>
<td>Breakout Board D (RoHS)</td>
<td>HVQFN16, 20 and 24</td>
<td></td>
</tr>
<tr>
<td>OM13495</td>
<td>Breakout Board E (RoHS)</td>
<td>Breakout Board E (RoHS)</td>
<td>TSSOP14, 16, 20 and 24</td>
<td></td>
</tr>
<tr>
<td>OM13497</td>
<td>Breakout Board F (RoHS)</td>
<td>Breakout Board F (RoHS)</td>
<td>QSOP16, XFBGA16, XQFN16, TSSOP23</td>
<td></td>
</tr>
<tr>
<td>OM13498</td>
<td>Breakout Board G (RoHS)</td>
<td>Breakout Board G (RoHS)</td>
<td>VFBS24, XFBGA24, HTSSOP28</td>
<td></td>
</tr>
</tbody>
</table>
Total Phase Debugging Tools

- Third party, industry leading, diagnostic tools
- Aardvark (I²C Host Adapter)

- Beagle (I²C Protocol Analyzer)

- Plugs in to the Fm+ Development Board
- Plugs in to the new Daughter Cards (allows standalone operation)
Aardvark (Master) and Beagle (Monitor)

- Third Party tools for I²C control and data logging
- Bundled software (for Win7/64, MAC, Linux)
- Not supplied by NXP (buy your own tools)

http://www.totalphase.com/products/beagle_ism/

https://www.totalphase.com/products/aardvark-i2cspi/
Key NXP Contacts

- **Product Marketing Director**
  - Giovanni Genna
  - Office: +41 (0)44 454 7767
  - Giovanni.Genna@nxp.com

- **Product Marketing Manager**
  - Steve Blozis
  - Office: 408-518-5384
  - Stephen.Blozis@nxp.com

- **Technical Marketing Manager**
  - Emmanuel T. Nana
  - Office: 408-518-5306
  - Emmanuel.Nana@nxp.com

- **Regional Marketing Manager**
  - Ravi Shah
  - Office: 408-518-5309
  - Ravi.Shah@nxp.com