DesignCon 2003
TecForum
I^{2}C Bus Overview
January 27 2003

Philips Semiconductors

Jean Marc Irazabal –Technical Marketing Manager for I^{2}C Devices

Steve Blozis –International Product Manager for I^{2}C Devices

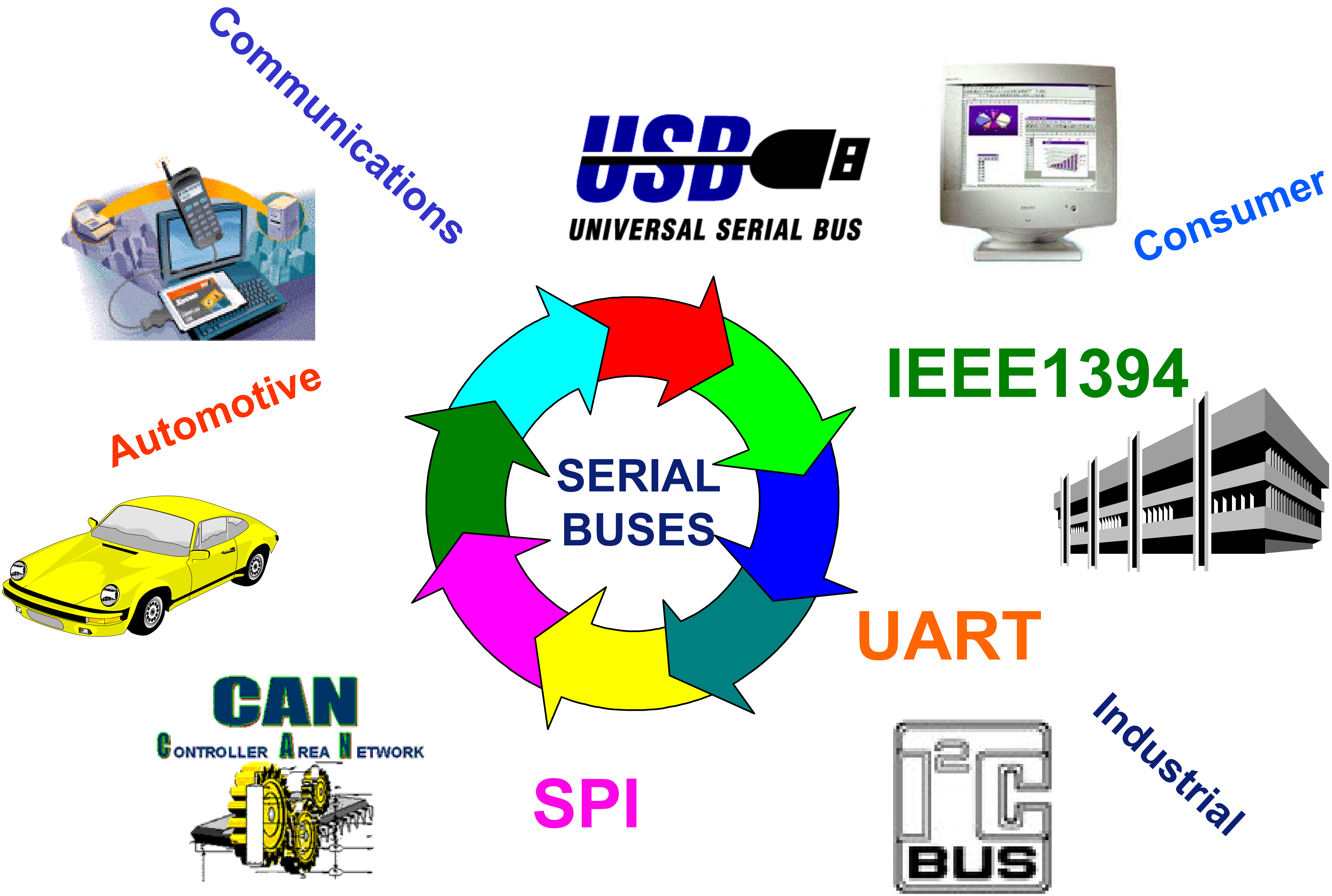
Agenda

- 1^{st} Hour
  - Serial Bus Overview
  - I^2C Theory Of Operation
- 2^{nd} Hour
  - Overcoming Previous Limitations
  - I^2C Development Tools and Evaluation Board
- 3^{rd} Hour
  - SMBus and IPMI Overview
  - I^2C Device Overview
  - I^2C Patent and Legal Information
  - Q & A

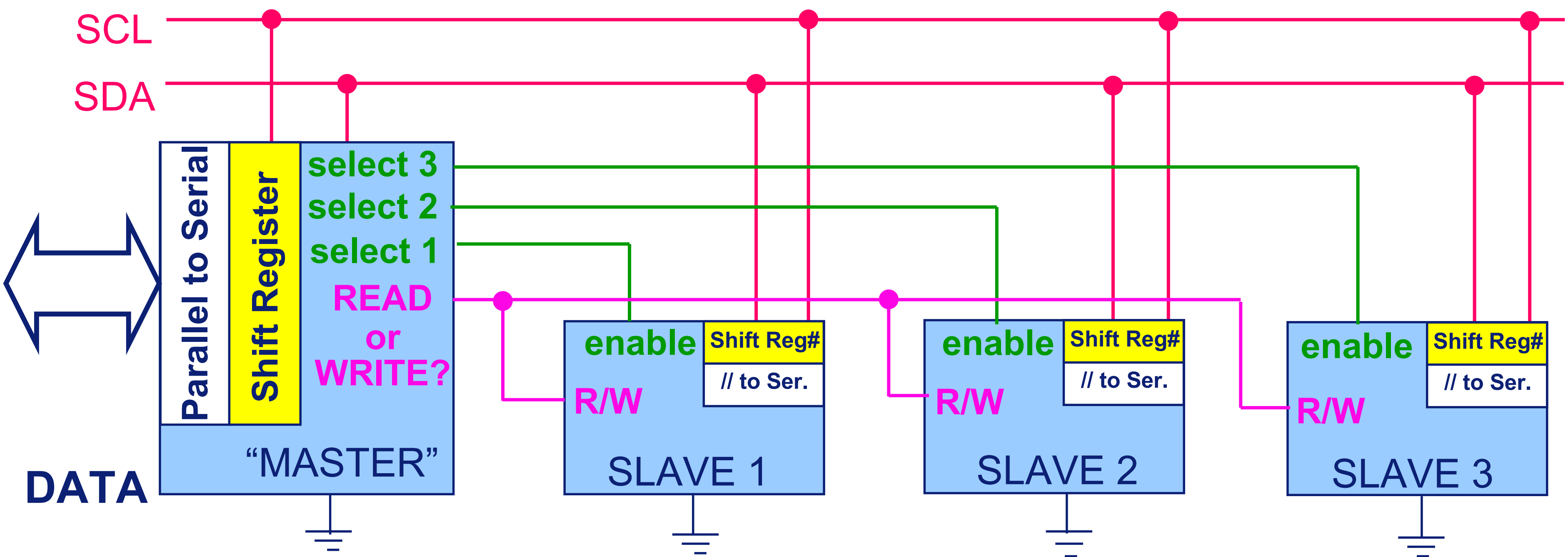
Slide speaker notes are included in AN10216 I^2C Manual

1st Hour

Serial Bus Overview

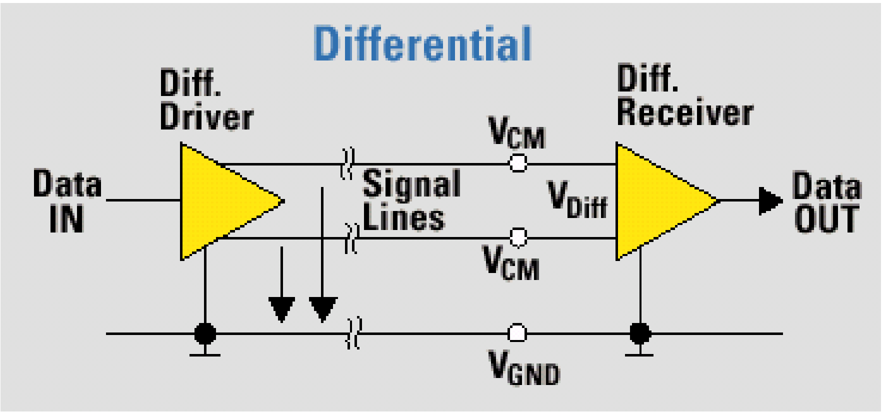

Communications
USB
UNIVERSAL SERIAL BUS
Consumer
IEEE1394
Automotive
SERIAL BUSES
UART
CAN
CONTROLLER AREA NETWORK
SPI
I2C
BUS
Industrial

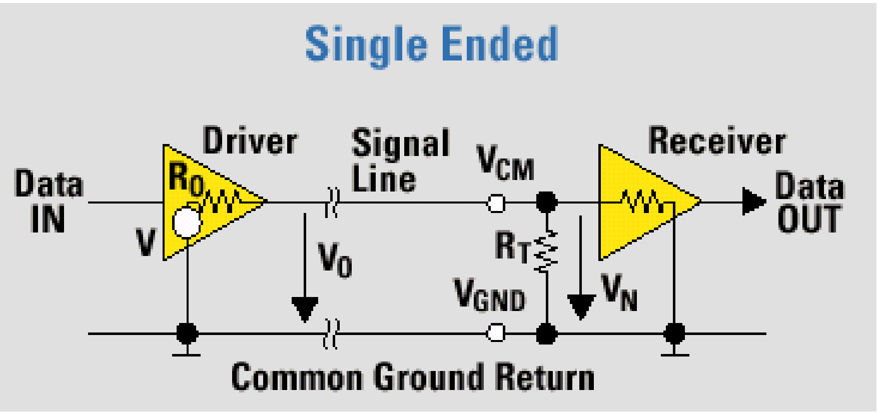
General concept for Serial communications

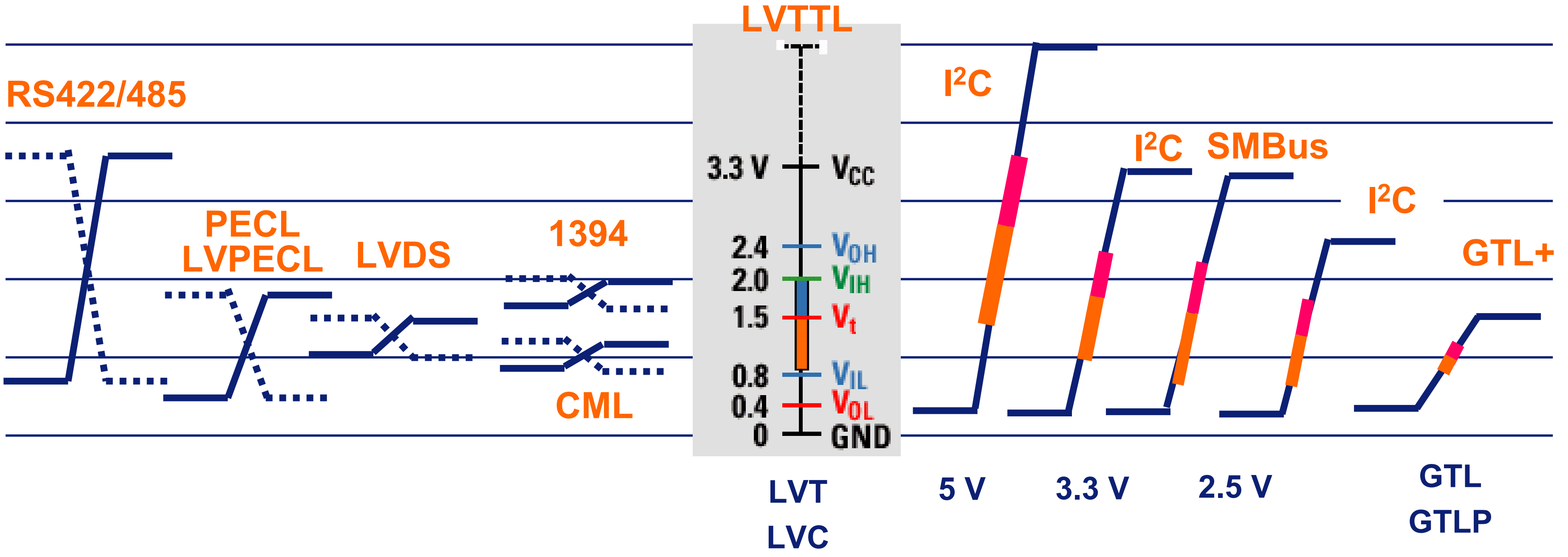


- A point to point communication does not require a Select control signal
- An asynchronous communication does not have a Clock signal
- Data, Select and R/W signals can share the same line, depending on the protocol
- Notice that Slave 1 cannot communicate with Slave 2 or 3 (except via the 'master')
  Only the 'master' can start communicating. Slaves can 'only speak when spoken to'

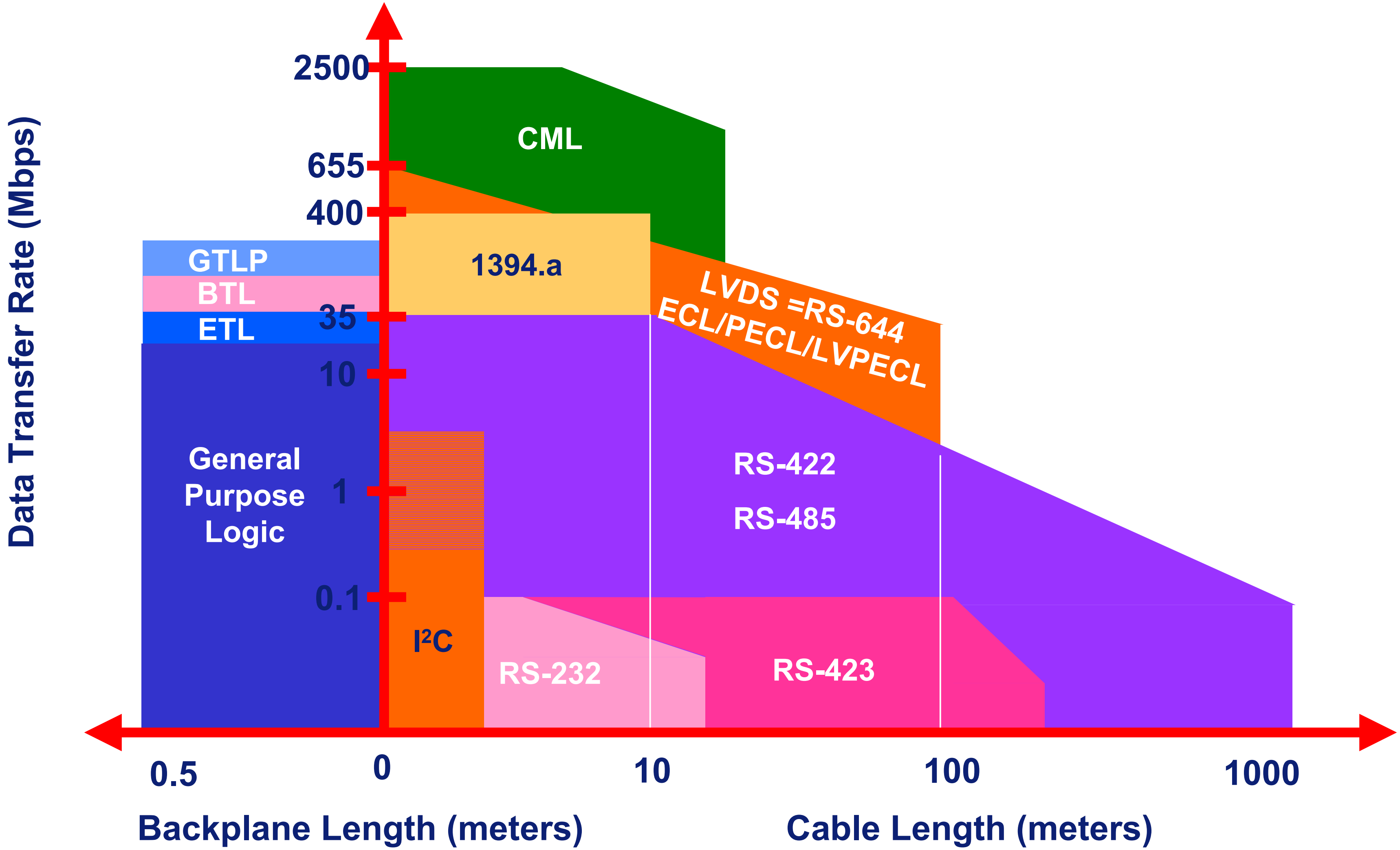
Typical Signaling Characteristics







Transmission Standards



Speed of various connectivity methods (bits/sec)

CAN (1 Wire)	**33 kHz (typ)**
I^2C ('Industrial', and SMBus)	**100 kHz**
SPI	**110 kHz (original speed)**
CAN (fault tolerant)	**125 kHz**
I^2C	**400 kHz**
CAN (high speed)	**1 MHz**
I^2C 'High Speed mode'	**3.4 MHz**
USB (1.1)	**1.5 MHz or 12 MHz**
SCSI (parallel bus)	**40 MHz**
Fast SCSI	**8-80 MHz**
Ultra SCSI-3	**18-160 MHz**
Firewire / IEEE1394	**400 MHz**
Hi-Speed USB (2.0)	**480 MHz**

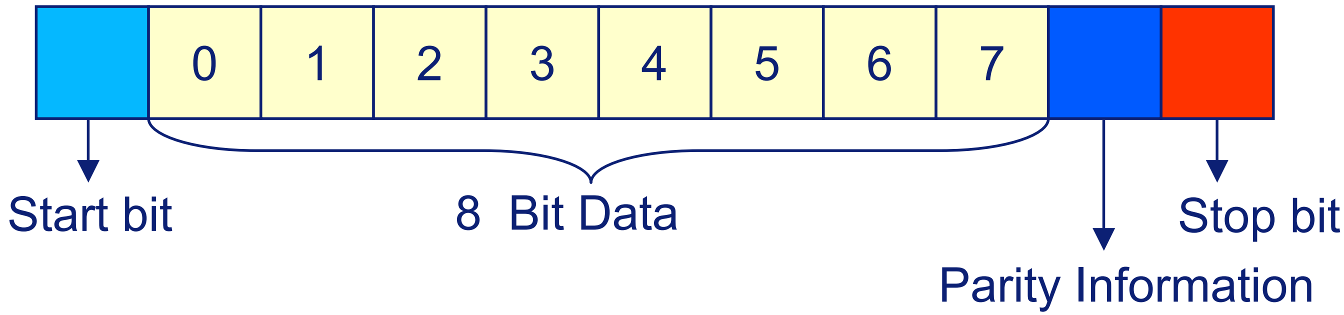
Bus characteristics compared

Bus	**Data rate** (bits / sec)	**Length** (meters)	Length limiting factor	**Nodes** Typ.number	Node number limiting factor
I^{2}C	**400k**	**2**	wiring capacitance	**20**	400pF max
I^{2}C with buffer	**400k**	**100**	propagation delays	**any**	no limit
I^{2}C high speed	**3.4M**	**0.5**	wiring capacitance	**5**	100pF max
CAN 1 wire	**33k**	**100**	total capacitance	**32**	load resistance and transceiver current drive
CAN differential	**5k**	**10km**	propagation delays	**100**	
	125k	**500**			
	1M	**40**			
USB (low-speed, 1.1)	**1.5M**	**3**	cable specs	**2**	bus specs
USB (full-speed, 1.1)	**1.5/12M**	**25**	5 cables linking 6 nodes (5m cable node to node)	**127**	bus and hub specs
Hi-Speed USB (2.0)	**480M**				
IEEE-1394	**100 to 400M+**	**72**	16 hops, 4.5M each	**63**	6-bit address

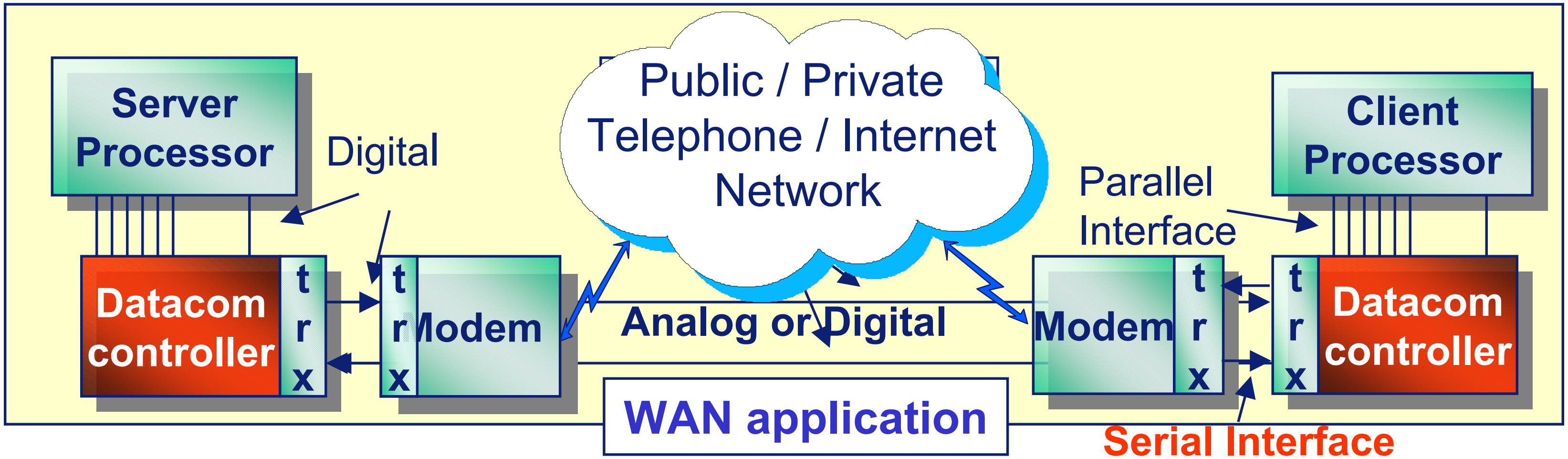
What is UART?

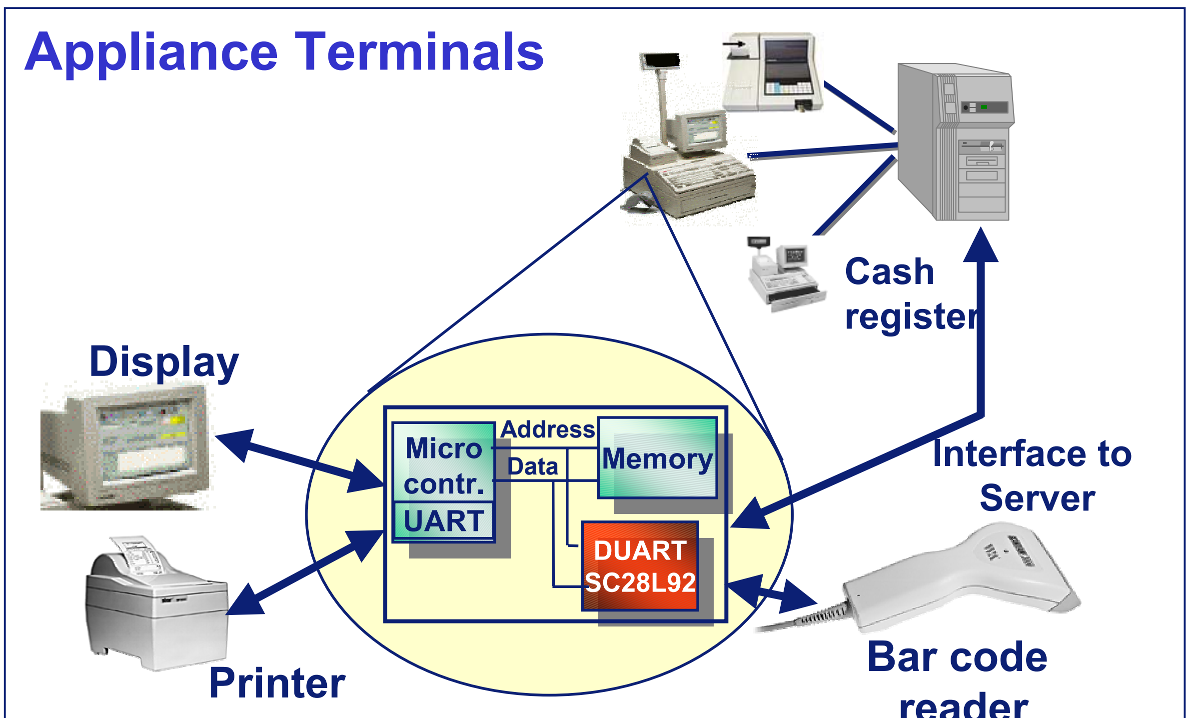
(Universal Asynchronous Receiver Transmitter)

- Communication standard implemented in the 60's.
- Simple, universal, well understood and well supported.
- Slow speed communication standard: up to 1 Mbits/s
- Asynchronous means that the data clock is not included in the data: Sender and Receiver must agree on timing parameters in advance.
- "Start" and "Stop" bits indicates the data to be sent
- Parity information can also be sent



UART - Applications



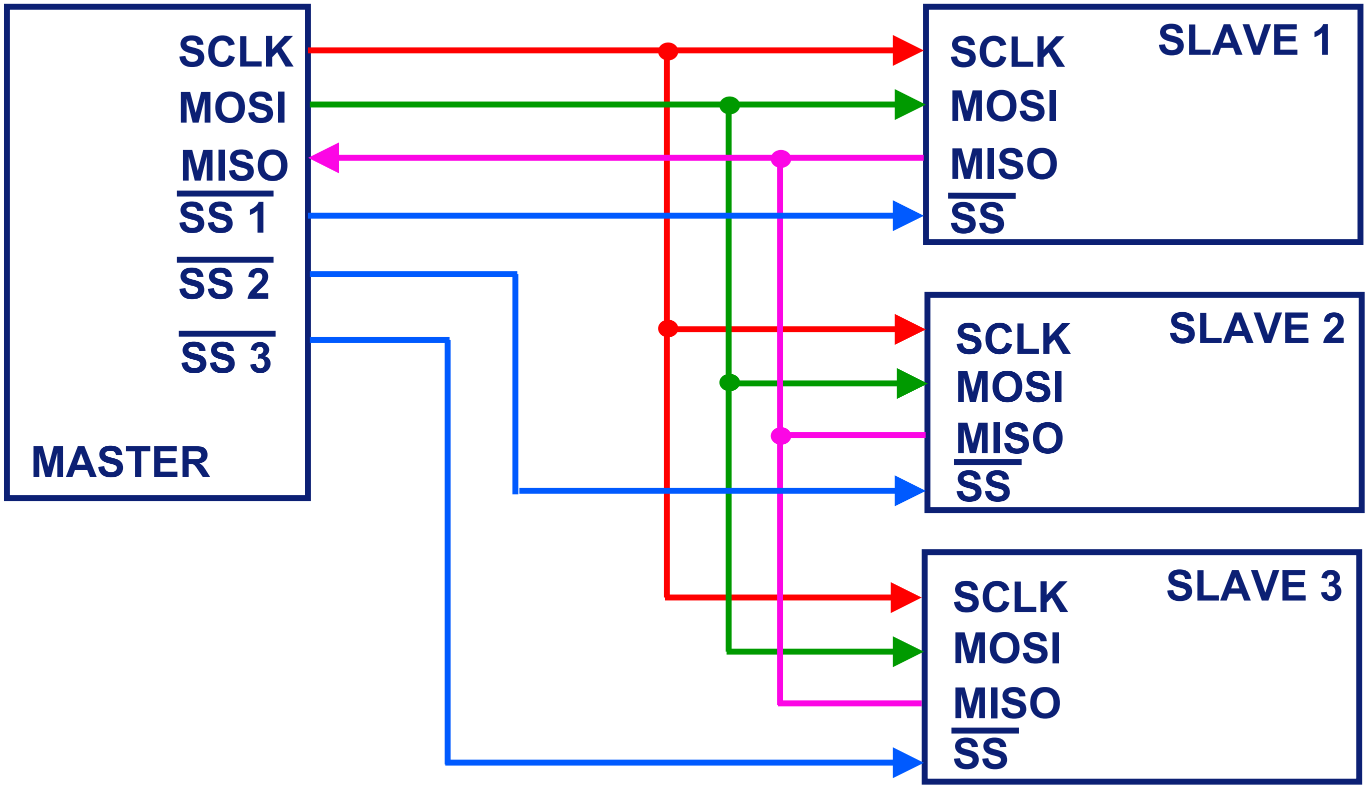


- Entertainment
- Home Security
- Robotics
- Automotive
- Cellular
- Medical

What is SPI?

- Serial Peripheral Interface (SPI) is a 4-wire full-duplex synchronous serial data link:
  - SCLK: Serial Clock
  - MOSI: Master Out Slave In - Data from Master to Slave
  - MISO: Master In Slave Out - Data from Slave to Master
  - SS: Slave Select
- Originally developed by Motorola
- Used for connecting peripherals to each other and to microprocessors
- Shift register that serially transmits data to other SPI devices
- Actually a “3 + n” wire interface with n = number of devices
- Only one master active at a time
- Various Speed transfers (function of the system clock)

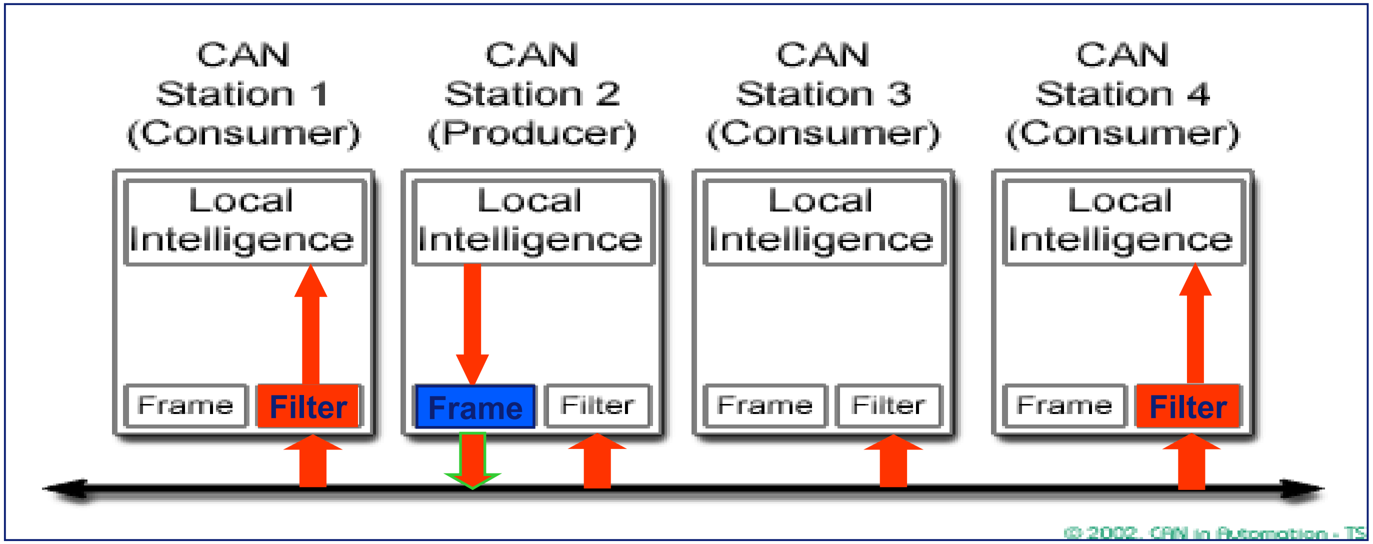
SPI - How are the connected devices recognized?



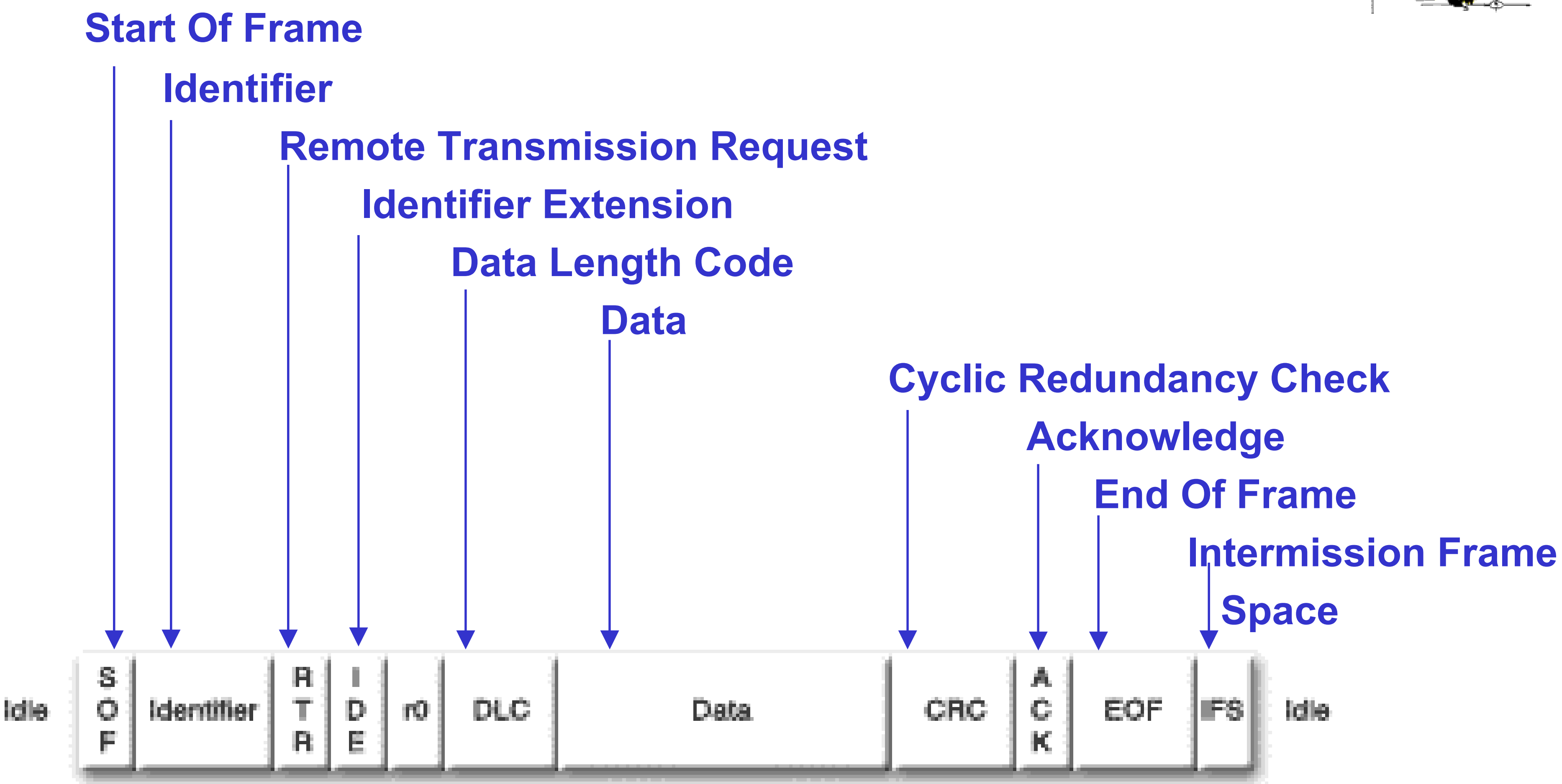
- Simple transfer scheme, 8 or 16 bits
- Allows many devices to use SPI through the addition of a shift register
- Full duplex communications
- Number of wires proportional to the number of devices in the bus

What is CAN ? (Controller Area Network)

- Proposed by Bosch with automotive applications in mind (and promoted by CIA - of Germany - for industrial applications)
- Relatively complex coding of the messages
- Relatively accurate and (usually) fixed timing
- All modules participate in every communication
- Content-oriented (message) addressing scheme



CAN protocol



- **Very intelligent controller requested to generate such protocol**

CAN Bus Advantages

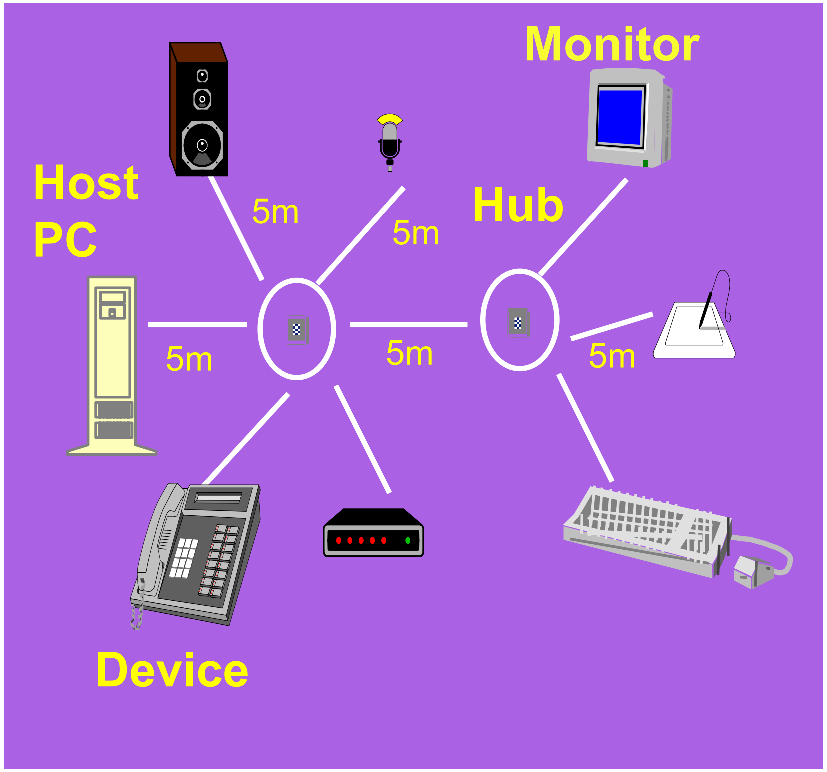
- Accepted standard for Automotive and industrial applications
  - interfacing between various vendors easier to implement
- Freedom to select suitable hardware
  - differential or 1 wire bus
- Secure communications, high Level of error detection
  - 15 bit CRC messages (Cyclic Redundancy Check)
  - Reporting / logging
  - Faulty devices can disconnect themselves
  - Low latency time
  - Configuration flexibility
- High degree of EMC immunity (when using Si-On-Insulator technology)

What is USB ? (Universal Serial Bus)

- Originally a standard for connecting PCs to peripherals
- Defined by Intel, Microsoft, …
- Intended to replace the large number of legacy ports in the PC
- Single master (= Host) system with up to 127 peripherals
- Simple plug and play; no need to open the PC
- Standardized plugs, ports, cables
- Has over 99% penetration on all new PCs
- Adapting to new requirements for flexibility of Host function
  - New Hardware/Software allows dynamic exchanging of Host/Slave roles
  - PC is no longer the only system Host. Can be a camera or a printer.

USB Topology (original concept, USB1.1, USB2.0)

- **Host**
  - One PC host per system
  - Provides power to peripherals
- **Hub**
  - Provides ports for connecting more peripheral devices.
  - Provides power, terminations
  - External supply or Bus Powered
- **Device, Interfaces and Endpoints**
  - Device is a collection of data interface(s)
  - Interface is a collection of endpoints (data channels)
  - Endpoint associated with FIFO(s) - for data I/O interfacing



USB Bus Advantages

- Hot pluggable, no need to open cabinets
- Automatic configuration
- Up to 127 devices can be connected together
- Push for USB to become THE standard on PCs
  – standard for iMac, supported by Windows, now on > 99%of PCs
- Interfaces (bridges) to other communication channels exist
  – USB to serial port (serial port vanishing from laptops)
  – USB to IrDA or to Ethernet
- Extreme volumes force down IC and hardware prices
- Protocol is evolving fast

Versions of USB specification

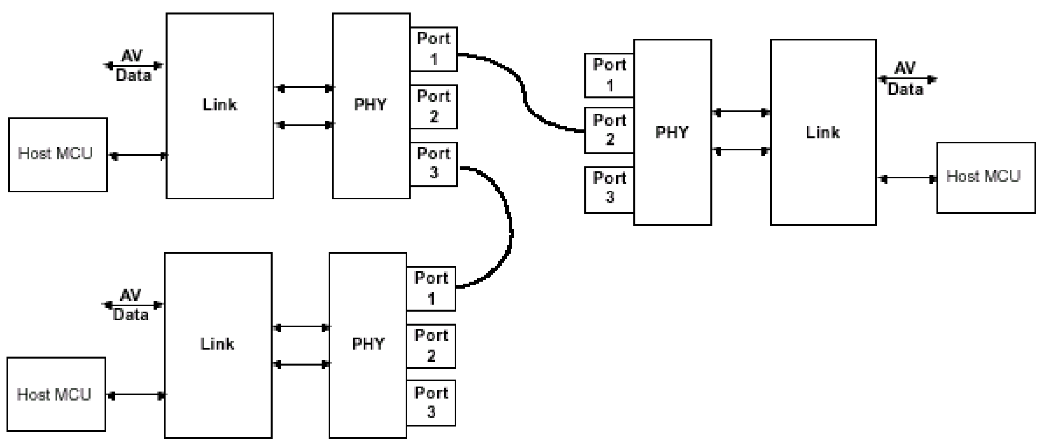
- **USB 1.1**
  - Established, large PC peripheral markets
  - Well controlled hardware, special 4-pin plugs/sockets
  - 12MBits/sec (normal) or 1.5Mbits/sec (low speed) data rate
- **USB 2.0**
  - Challenging IEEE1394/Firewire for video possibilities
  - 480 MHz clock for Hi-Speed means it's real "UHF" transmission
  - Hi-Speed option needs more complex chip hardware and software
  - Hi-Speed component prices about x 2 compared to full speed
- **USB "OTG" (On The Go) Supplement**
  - New hardware - smaller 5-pin plugs/sockets
  - Lower power (reduced or no bus-powering)

What is IEEE1394 ?

- A bus standard devised to handle the high data throughput requirements of MPEG-2 and DVD
  - Video requires constant transfer rates with guaranteed bandwidth
  - Data rates 100, 200, 400 Mbits/sec and looking to 3.2 Gb/s
- Also known as "Firewire" bus (registered trademark of Apple)
- Automatically re-configures itself as each device is added
  - True plug & play
  - Hot-plugging of devices allowed
- Up to 63 devices, 4.5 m cable 'hops', with max. 16 hops
- Bandwidth guaranteed

1394 Topology

Multiple Nodes interconnected with a multiple twisted-pair cable



- Physical layer
  - Analog interface to the cable
  - Simple repeater
  - Performs bus arbitration
- Link layer
  - Assembles and dis-assembles bus packets
  - Handles response and acknowledgment functions
- Host controller
  - Implements higher levels of the protocol

What is I²C ? (Inter-IC)

- Originally, bus defined by Philips providing a simple way to talk between IC's by using a minimum number of pins
- A set of specifications to build a simple universal bus guaranteeing compatibility of parts (ICs) from different manufacturers:
  - Simple Hardware standards
  - Simple Software protocol standard
- No specific wiring or connectors - most often it's just PCB tracks
- Has become a recognised standard throughout our industry and is used now by ALL major IC manufacturers

I^2C Bus - Software

- Simple procedures that allow communication to start, to achieve data transfer, and to stop
  - Described in the Philips protocol (rules)
  - Message serial data format is very simple
  - Often generated by simple software in general purpose micro
  - Dedicated peripheral devices contain a complete interface
  - Multi-master capable with arbitration feature
- Each IC on the bus is identified by its own address code
  - Address has to be unique
- The master IC that initiates communication provides the clock signal (SCL)
  - There is a maximum clock frequency but NO MINIMUM SPEED

How are the connected devices recognized?

- Master device ‘polls’ used a specific unique identification or “addresses” that the designer has included in the system
- Devices with Master capability can identify themselves to other specific Master devices and advise their own specific address and functionality
  - Allows designers to build ‘plug and play’ systems
  - Bus speed can be different for each device, only a maximum limit
- Only two devices exchange data during one ‘conversation’

Pros and Cons of the different buses

UART	CAN	USB	SPI	I^2C
• Well Known • Cost effective • Simple	• Secure • Fast	• Fast • Plug&Play HW • Simple • Low cost	• Fast • Universally accepted • Low cost • Large Portfolio	• Simple • Well known • Universally accepted • Plug&Play • Large portfolio • Cost effective
• Limited functionality • Point to Point	• Complex • Automotive oriented • Limited portfolio • Expensive firmware	• Powerful master required • No Plug&Play SW - Specific drivers required	• No Plug&Play HW • No “fixed” standard	• Limited speed

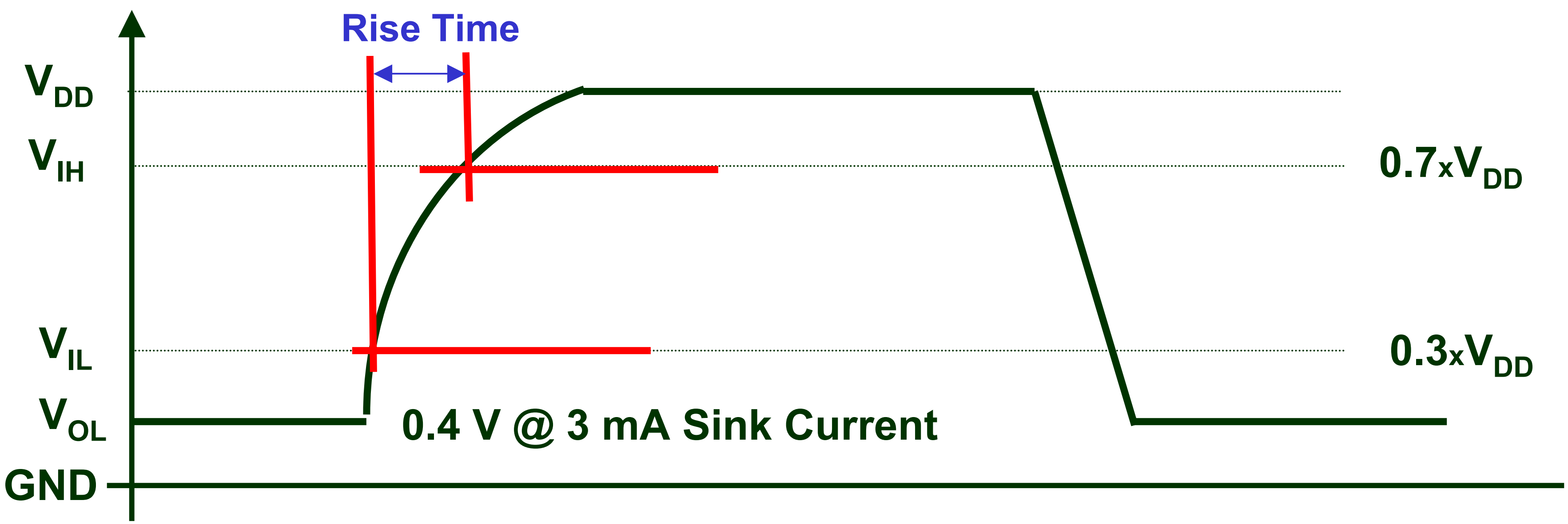
I^2C Theory Of Operation

I^2C Introduction

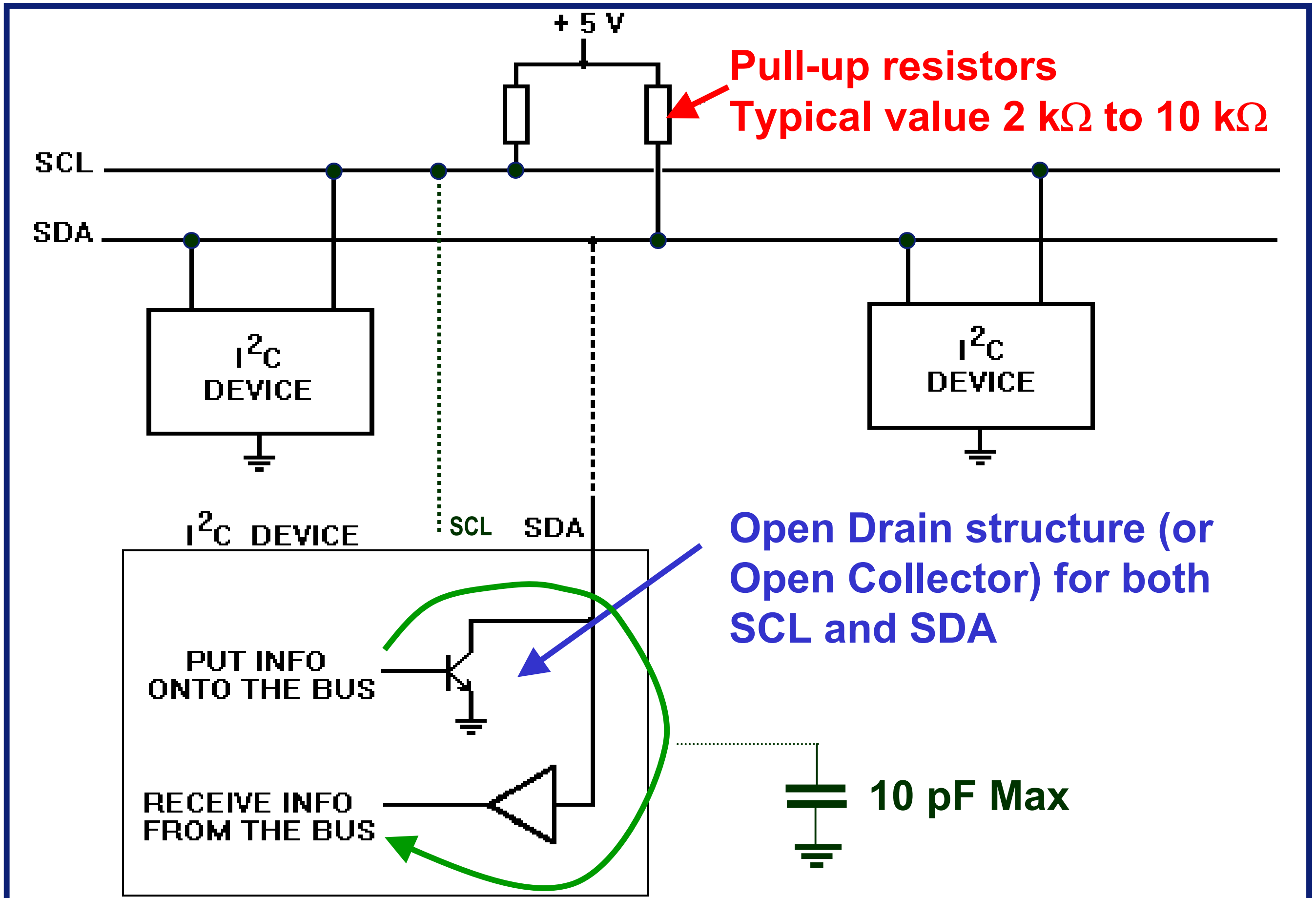
- I^2C bus = Inter-IC bus
- Bus developed by Philips in the 80's
- Simple bi-directional 2-wire bus:
  - serial data (SDA)
  - serial clock (SCL)
- Has become a worldwide industry standard and used by all major IC manufacturers
- Multi-master capable bus with arbitration feature
- Master-Slave communication; Two-device only communication
- Each IC on the bus is identified by its own address code
- The slave can be a:
  - receiver-only device
  - transmitter with the capability to both receive and send data

I^2C by the numbers

	Standard-Mode	Fast-Mode	High-Speed-Mode	
Bit Rate (kbits/s)	0 to 100	0 to 400	0 to 1700	0 to 3400
Max Cap Load (pF)	400	400	400	100
Rise time (ns)	1000	300	160	80
Spike Filtered (ns)	N/A	50	10	
Address Bits	7 and 10	7 and 10	7 and 10	

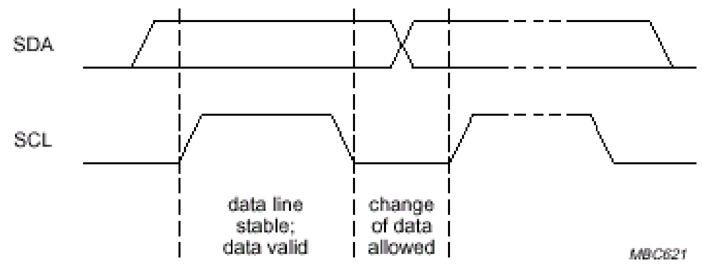


I²C Hardware architecture



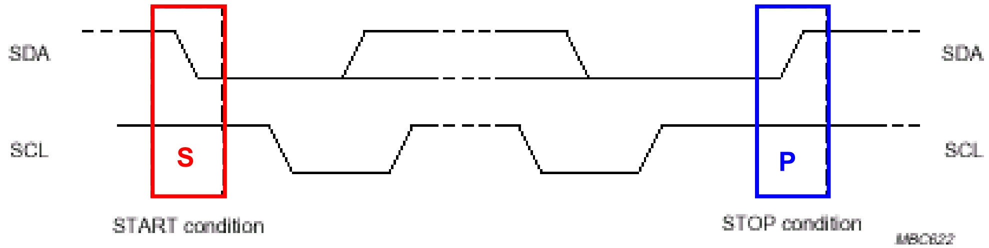
START/STOP conditions

- **Data on SDA must be stable when SCL is High**



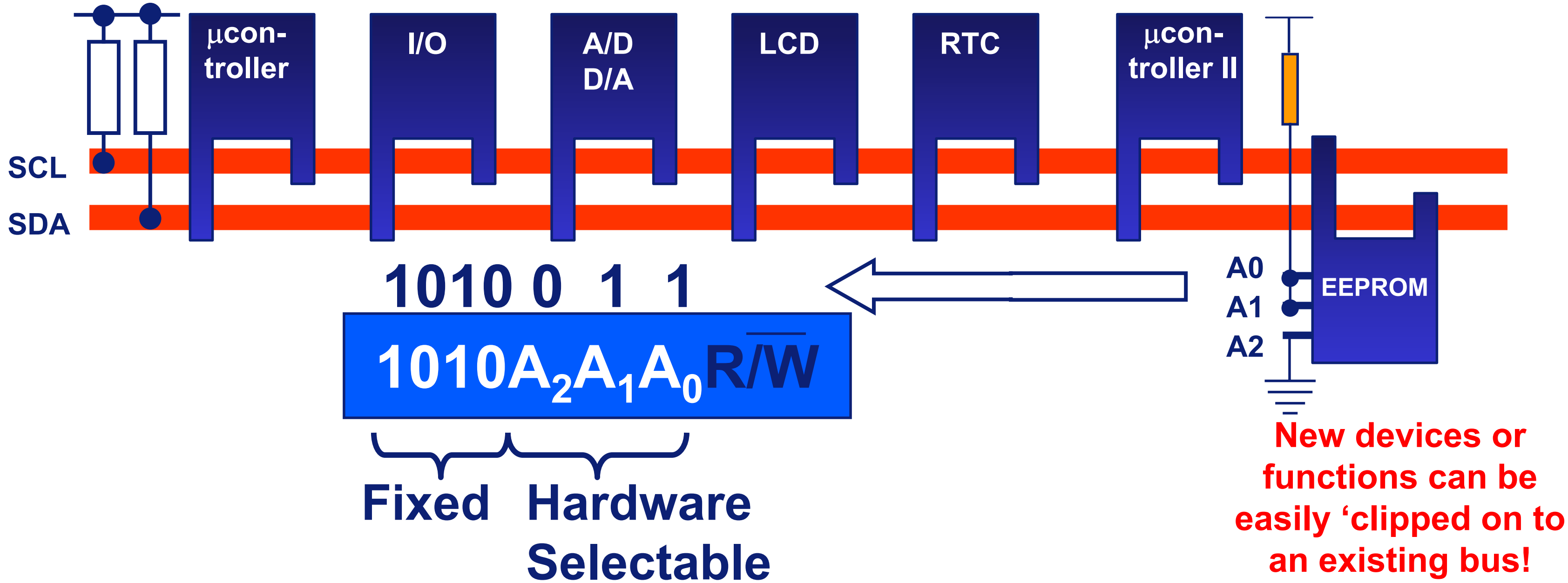
Bit transfer on the I^2C-bus.

- **Exceptions are the START and STOP conditions**



START and STOP conditions.

I^{2}C Address, Basics

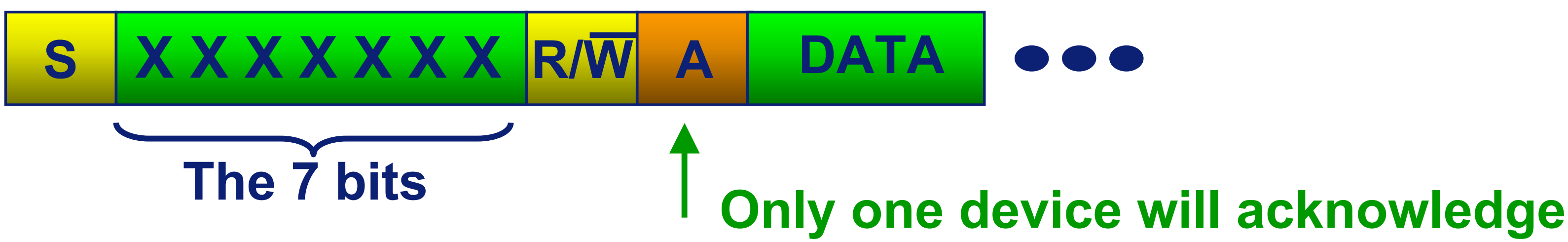


- Each device is addressed individually by software
- Unique address per device: fully fixed or with a programmable part through hardware pin(s).
- Programmable pins mean that several same devices can share the same bus
- Address allocation coordinated by the I^{2}C-bus committee
- 112 different types of devices max with the 7-bit format (others reserved)

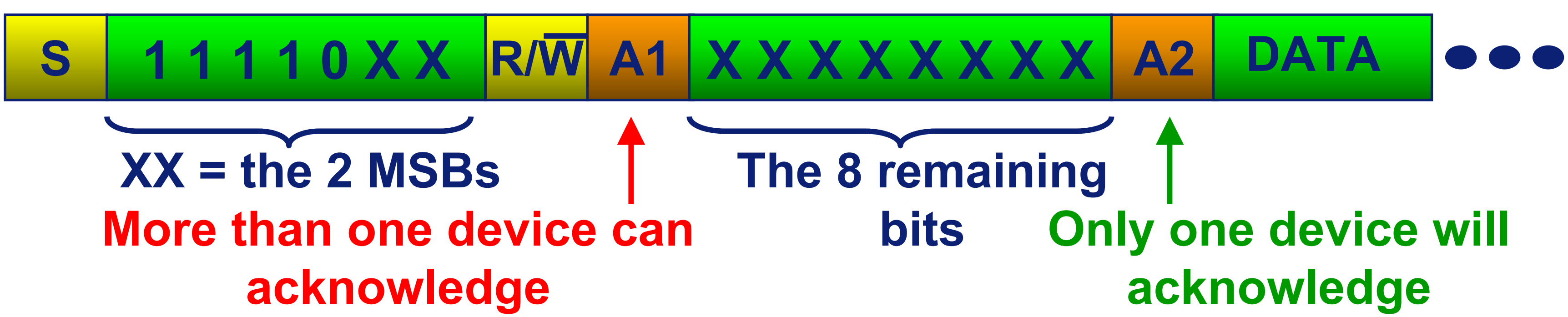
I^{2}C Address, 7-bit and 10-bit formats

- The 1st byte after START determines the Slave to be addressed
- Some exceptions to the rule:
  - "General Call" address: all devices are addressed : 0000 000 + R/W = 0
  - 10-bit slave addressing : 1111 0XX + R/W = X

•7-bit addressing



• 10-bit addressing

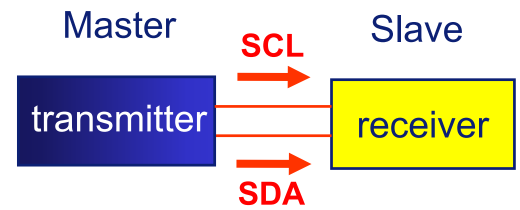


I^{2}C Read and Write Operations (1)

- **Write to a Slave device**

				< n data bytes >				
S	slave address	$\overline{W}$	A	data	A	data	A	P

"0" = Write **Each byte is acknowledged by the slave device**

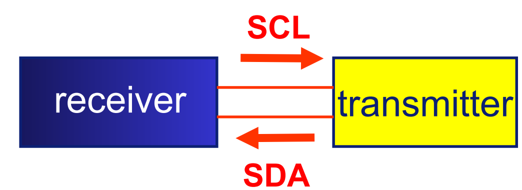


The master is a "MASTER - TRANSMITTER":
–it transmits both Clock and Data during the all communication

- **Read from a Slave device**

				< n data bytes >				
S	slave address	R	A	data	A	data	$\overline{A}$	P

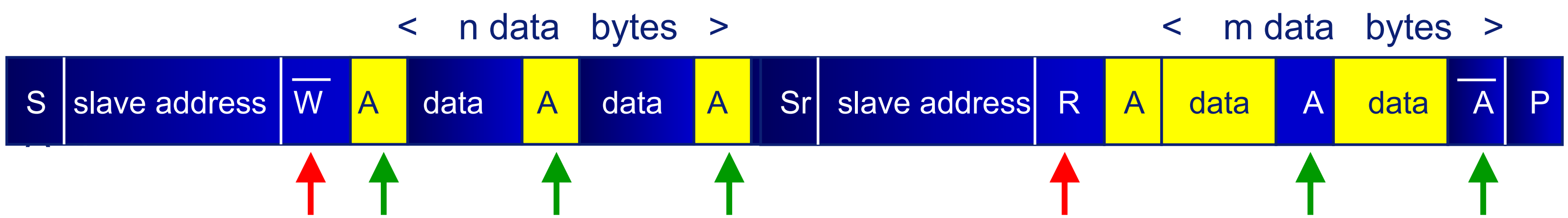
"1" = Read **Each byte is acknowledged by the master device (except the last one, just before the STOP condition)**



The master is a "MASTER TRANSMITTER then MASTER - RECEIVER":
– it transmits Clock all the time
– it sends slave address data and then becomes a receiver

I²C Read and Write Operations (2)

- **Combined Write and Read**



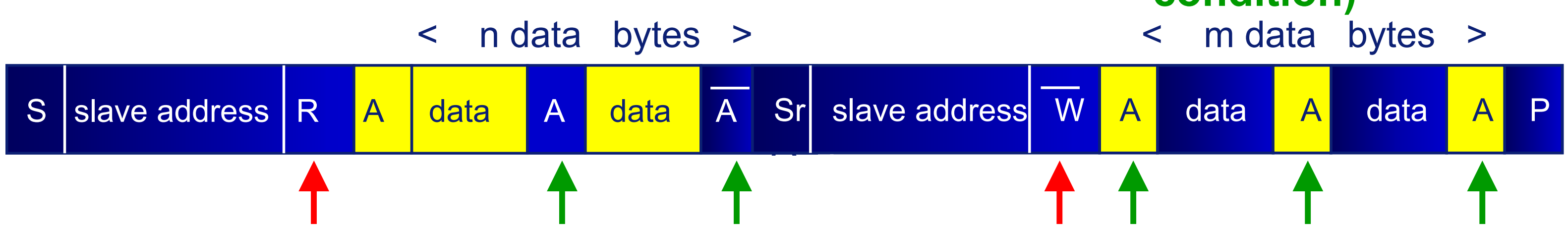
"0" = Write

Each byte is acknowledged by the slave device

"1" = Read

Each byte is acknowledged by the master device (except the last one, just before the STOP condition)

- **Combined Read and Write**



"1" = Read

Each byte is acknowledged by the master device (except the last one, just before the Re-START condition)

"0" = Write

Each byte is acknowledged by the slave device

Acknowledge; Clock Stretching

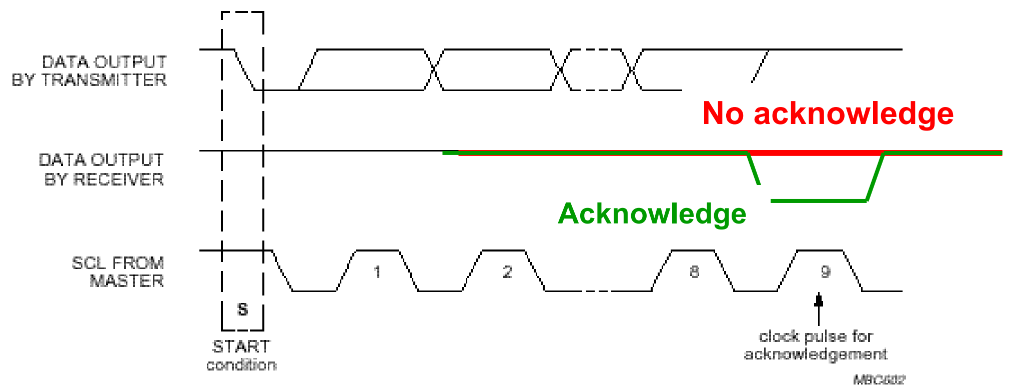
- **Acknowledge**

  Done on the 9th clock pulse and is mandatory

  → Transmitter releases the SDA line

  → Receiver pulls down the SDA line (SCL must be HIGH)

  → Transfer is aborted if no acknowledge

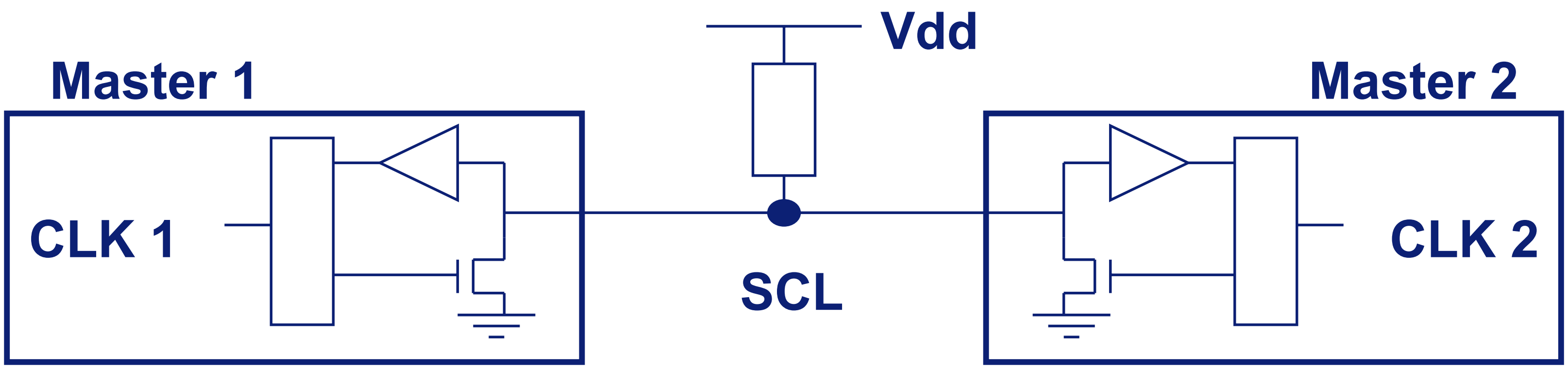


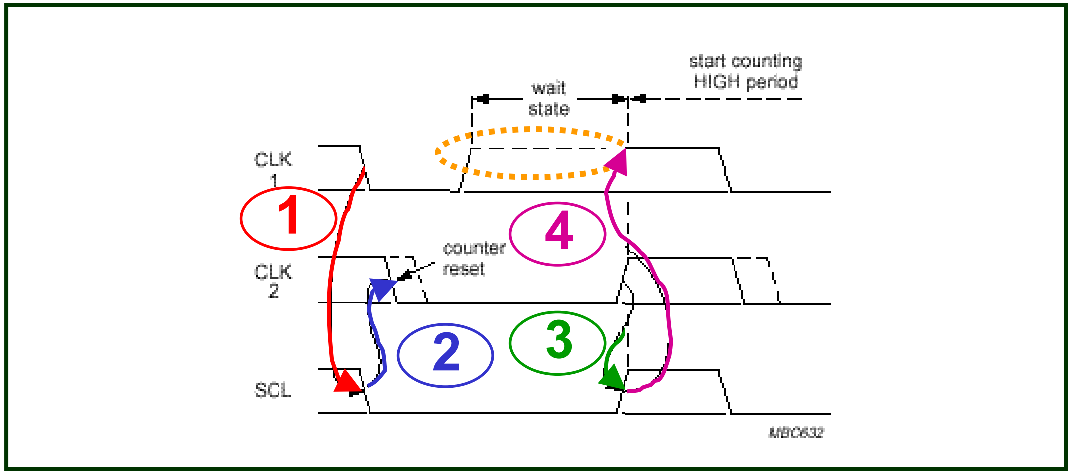
Acknowledge on the I²C-bus.

- **Clock Stretching**

  - Slave device can hold the CLOCK line LOW when performing other functions
  - Master can slow down the clock to accommodate slow slaves

I^2C Protocol - Clock Synchronization

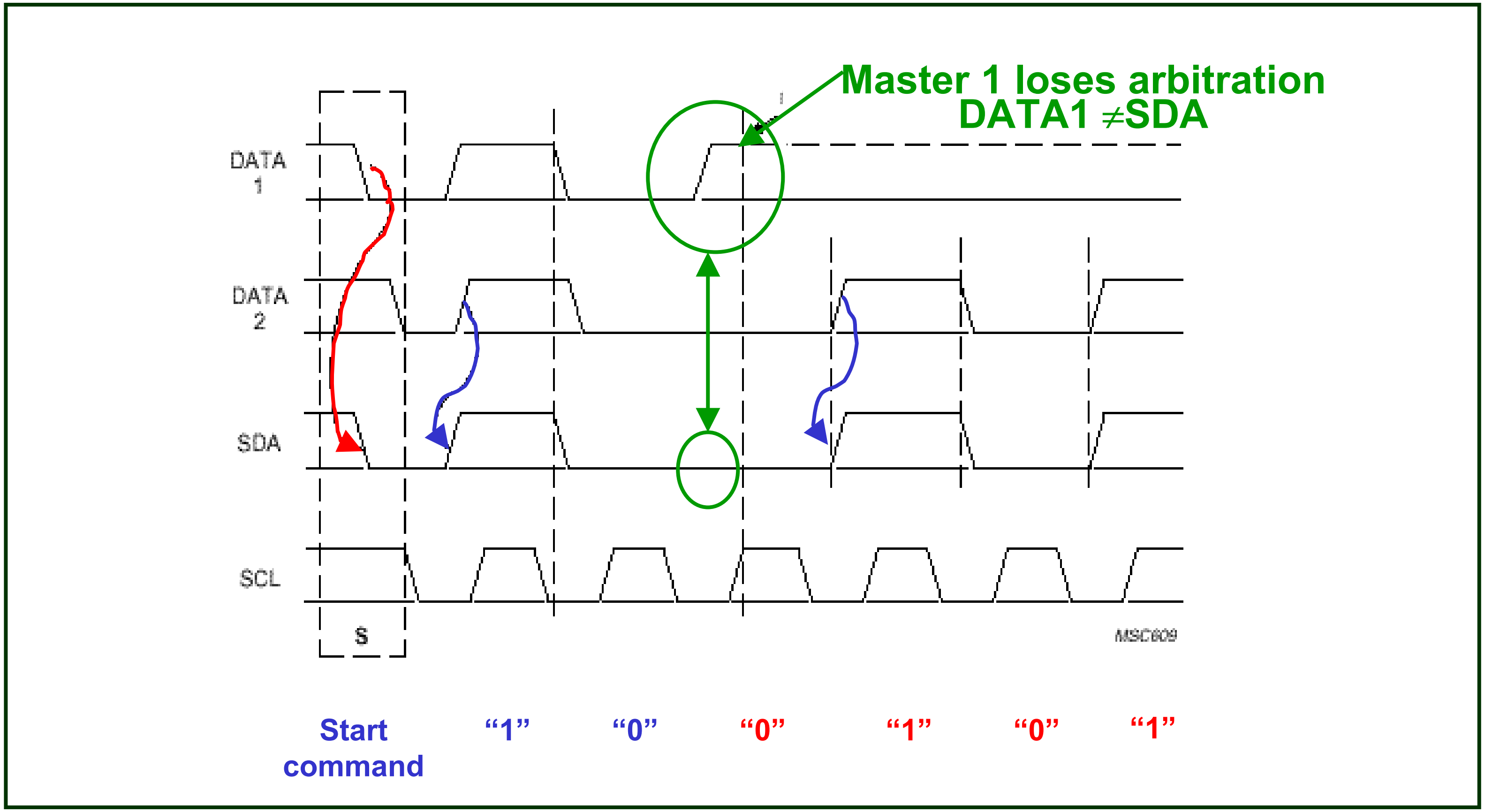




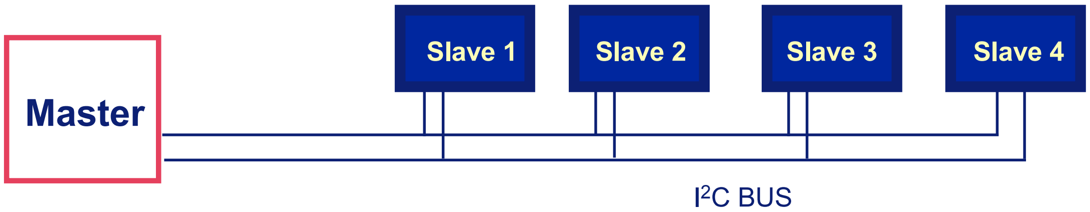
- **LOW** period determined by the **longest clock LOW** period
- **HIGH** period determined by **shortest clock HIGH** period

I^{2}C Protocol - Arbitration

- Two or more masters may generate a START condition at the same time
- Arbitration is done on SDA while SCL is HIGH - Slaves are not involved



What do I need to drive the I^{2}C bus?



There are 3 basic ways to drive the I^{2}C bus:

1) With a Microcontroller with on-chip I^{2}C Interface
Bit oriented - CPU is interrupted after every bit transmission (Example: 87LPC76x)
Byte oriented - CPU can be interrupted after every byte transmission (Example: 87C552)

2) With ANY microcontroller: 'Bit Banging'
The I^{2}C protocol can be emulated bit by bit via any bi-directional open drain port

3) With a microcontroller in conjunction with bus controller like the PCF8584 or PCA9564 parallel to I^{2}C bus interface IC

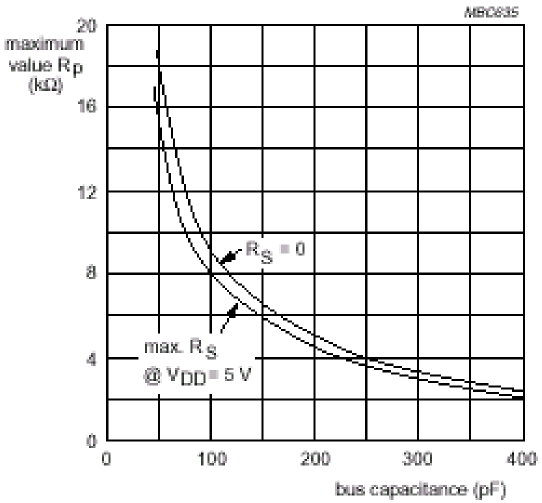
Pull-up Resistor calculation

DC Approach - Static Load

Worst Case scenario: maximum current load that the output transistor can handle → 3 mA . This gives us the minimum pull-up resistor value

$$R = \frac{\text{Vdd min} - 0.4\ \text{V}}{3\ \text{mA}}$$

With Vdd = 5V (min 4.5 V), Rmin = 1.3 kΩ



AC Approach - Dynamic load

- maximum value of the rise time:
  - 1µs for Standard-mode (100 kHz)
  - 0.3 µs for Fast-mode (400 kHz)
- Dynamic load is defined by:
  - device output capacitances (number of devices)
  - trace, wiring

$V(t) = V_{DD}\,(1-e^{-t/RC})$

Rising time defined between 30% and 70%

$T_{rise} = 0.847.RC$

I^2C Bus recovery

- Typical case is when masters fails when doing a read operation in a slave
- SDA line is then non usable anymore because of the “Slave-Transmitter” mode.
- Methods to recover the SDA line are:
  – Reset the slave device (assuming the device has a Reset pin)
  – Use a bus recovery sequence to leave the “Slave-Transmitter” mode
- Bus recovery sequence is done as following:

  1 - Send 9 clock pulses on SCL line

  2 - Ask the master to keep SDA High until the “Slave-Transmitter” releases the SDA line to perform the ACK operation

  3 - Keeping SDA High during the ACK means that the “Master-Receiver” does not acknowledge the previous byte receive

  4 - The “Slave-Transmitter” then goes in an idle state

  5 - The master then sends a STOP command initializing completely the bus

I²C Protocol Summary

START	HIGH to LOW transition on SDA while SCL is HIGH
STOP	LOW to HIGH transition on SDA while SCL is HIGH
DATA	8-bit word, MSB first (Address, Control, Data) - must be stable when SCL is HIGH - can change only when SCL is LOW - number of bytes transmitted is unrestricted
ACKNOWLEDGE	- done on each 9th clock pulse during the HIGH period - the transmitter releases the bus - SDA HIGH - the receiver pulls DOWN the bus line - SDA LOW
CLOCK	- Generated by the master(s) - Maximum speed specified but NO minimum speed - A receiver can hold SCL LOW when performing another function (transmitter in a Wait state) - A master can slow down the clock for slow devices
ARBITRATION	- Master can start a transfer only if the bus is free - Several masters can start a transfer at the same time - Arbitration is done on SDA line - Master that lost the arbitration must stop sending data

I^2C Summary - Advantages

- Simple Hardware standard
- Simple protocol standard
- Easy to add / remove functions or devices (hardware and software)
- Easy to upgrade applications
- Simpler PCB: Only 2 traces required to communicate between devices
- Very convenient for monitoring applications
- Fast enough for all “Human Interfaces” applications
  - Displays, Switches, Keyboards
  - Control, Alarm systems
- Large number of different I^2C devices in the semiconductors business
- Well known and robust bus

2^{nd} Hour

Overcoming Previous Limitations

How to solve I²C address conflicts?

- I²C protocol limitation: when a device does not have its I²C address programmable (fixed), only one same device can be plugged in the same bus

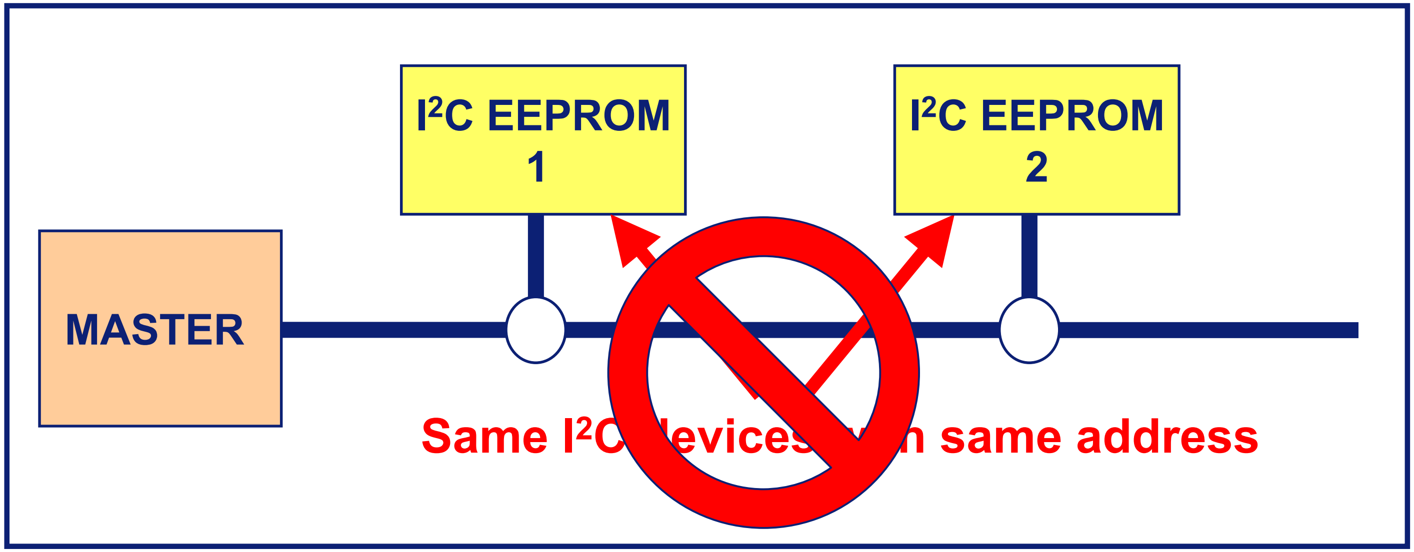
➔ An I²C multiplexer can be used to get rid of this limitation

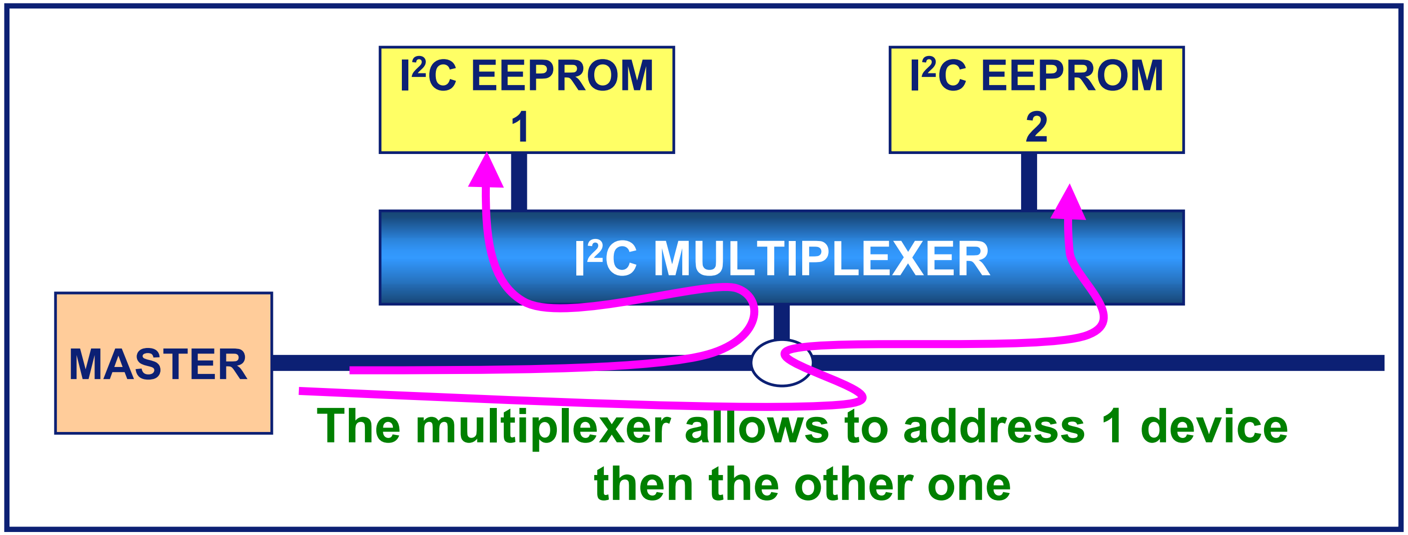
- It allows to split dynamically the main I²C in several sub-branches in order to talk to one device at a time
- It is programmable through I²C so no additional pins are required for control
- More than one multiplexer can be plugged in the same I²C bus

- Products

# of Channels	Standard	w/Interrupt Logic
2	PCA9540	PCA9542/43
4	PCA9546	PCA9544/45
8	PCA9548	

I²C Multiplexers: Address Deconflict





How to go beyond I²C max cap load?

• I²C protocol limitation: the maximum capacitive load in a bus is 400 pF. If the load is higher AC parameters will be violated.

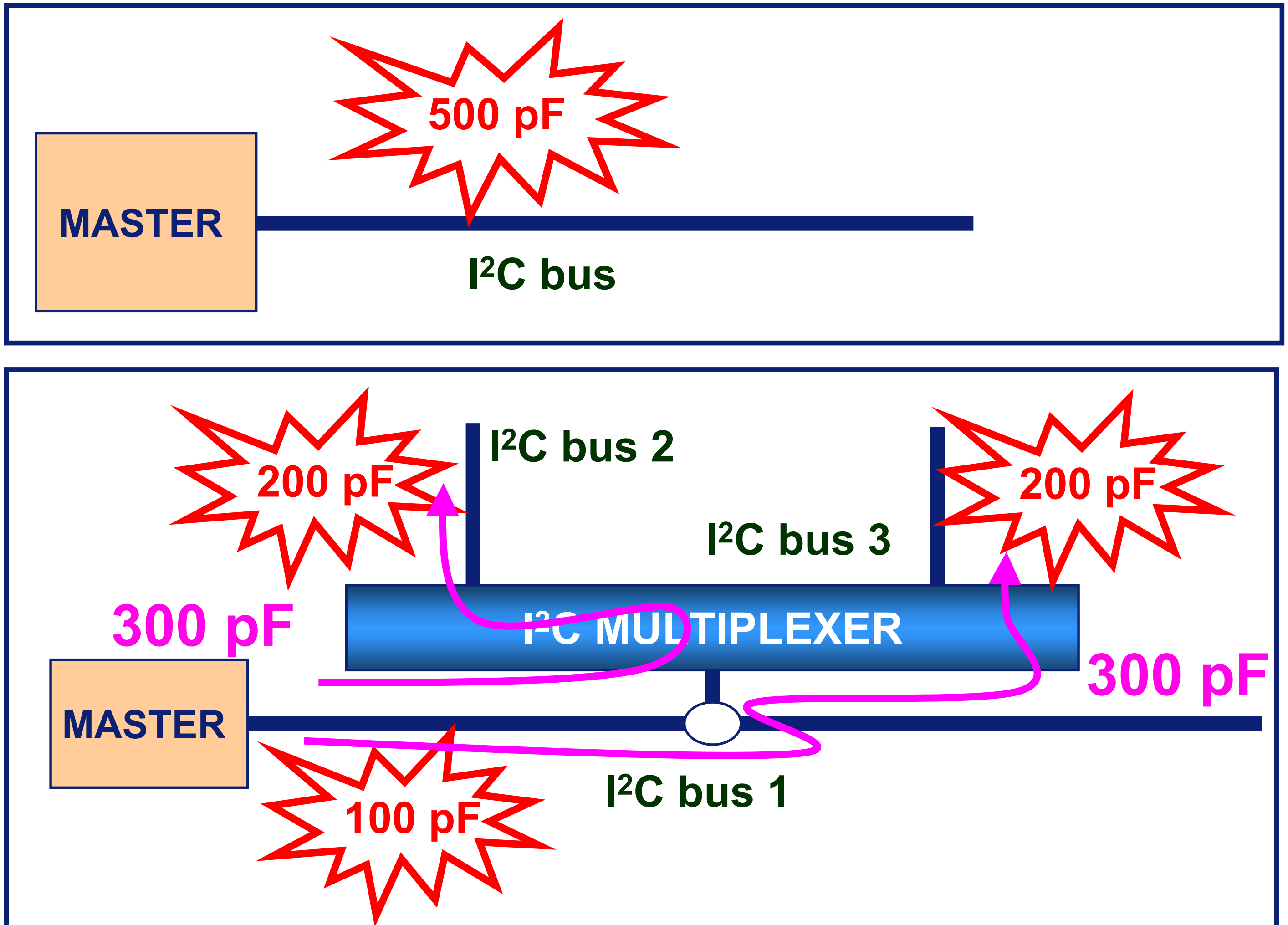
➔ An I²C multiplexer can be used to get rid of this limitation

• It allows to split dynamically the main I²C in several sub-branches in order to divide the bus capacitive load
• It is programmable through I²C so no additional pins are required for control
• More than one multiplexer can be plugged in the same I²C bus
• LIMITATION: All the sub-branches cannot be addressed at the same time

• Products:

# of Channels	Standard	w/Interrupt Logic
2	PCA9540	PCA9542/43
4	PCA9546	PCA9544/45
8	PCA9548	

I²C Multiplexers: Capacitive load split



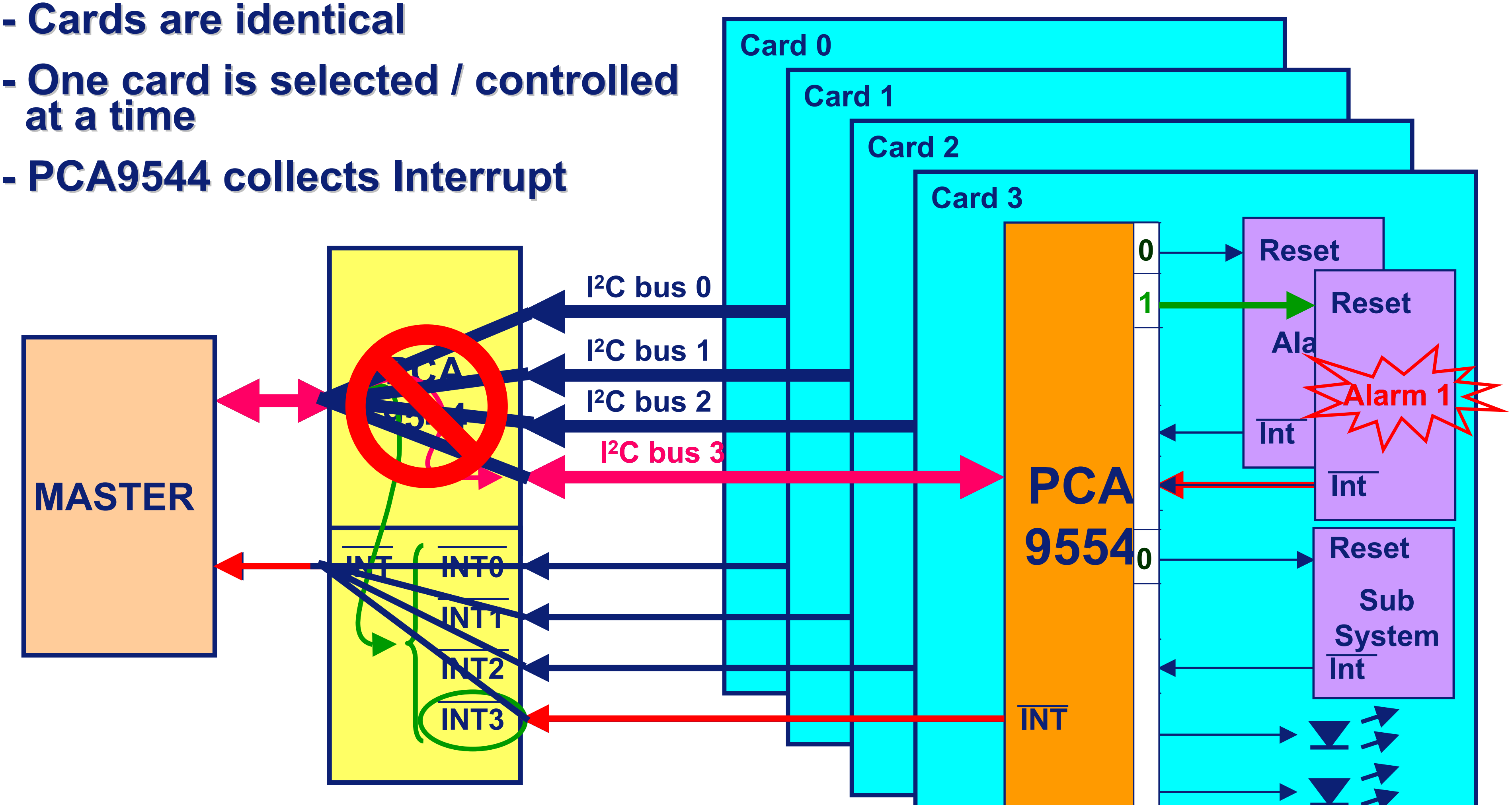
The multiplexer splits the bus in two downstream 200 pF busses + 100 pF upstream

Practical case: Multi-card application

- The following example shows how to build an application where:
  - Four identical control cards are used (same devices, same I^2Caddress)
  - Devices in each card are controlled through I^2C
  - Each card monitors and controls some digital information
  - Digital information is:
    1) Interrupt signals (Alarm monitoring)
    2) Reset signals (device initialization, Alarm Reset)
  - Each card generates an Interrupt when one (or more) device generates an Interrupt (Alarm condition detected)
  - The master can handle only one Interrupt signal for all the application

I^2C Multiplexers: Multi-card Application

- Cards are identical
- One card is selected / controlled at a time
- PCA9544 collects Interrupt



Interrupt signals are collected into one signal

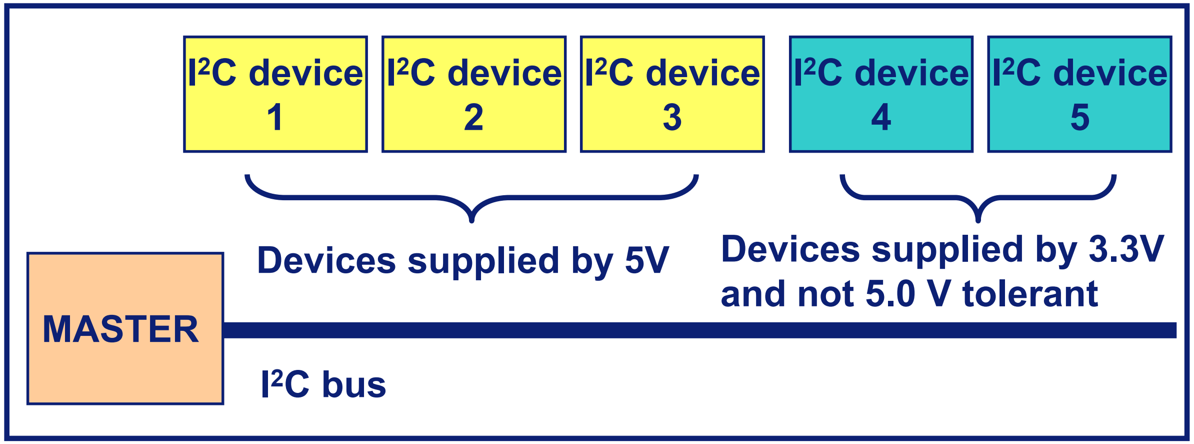
How to accommodate different I^2C logic levels in the same bus?

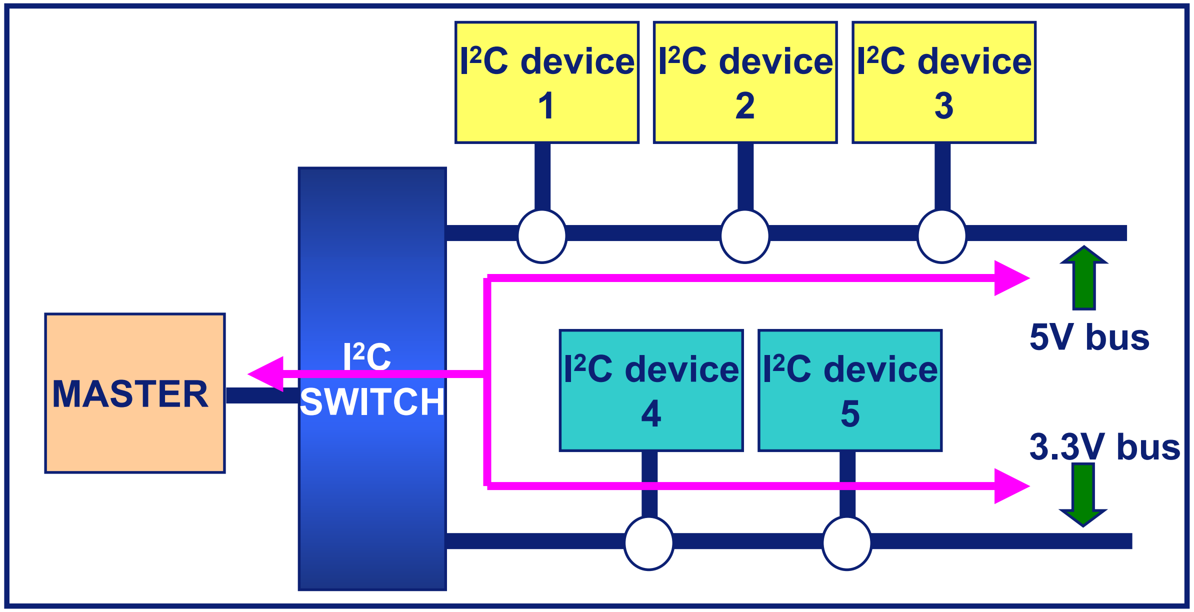
• I^2C protocol: Due to the open drain structure of the bus, voltage level in the bus is fixed by the voltage connected to the pull-up resistor. If different voltage levels are required (e.g., master core at 1.8 V, legacy I^2C bus at 5 V and new devices at 3.3 V), voltage level translators need to be used

➔ **An I^2C switch can be used to accommodate those different voltage levels.**

• It allows to split dynamically the main I^2C in several sub-branches and allow different supply voltages to be connected to the pull up resistors
• PCA devices are programmable through I^2C bus so no additional pin is required to control which channel is active
• More than one channel can be active at the same time so the master does not have to remember which branch it has to address (broadcast)
• More than one switch can be plugged in the same I^2C bus

I²C Switches: Voltage Level Shifting





- Products

# Channels		Int
1	GTL2002	
2	PCA9540	
	PCA9542/43	X
4	PCA9546	
	PCA9544/45	X
5	GTL2010	
8	PCA9548	
11	GTL2000	

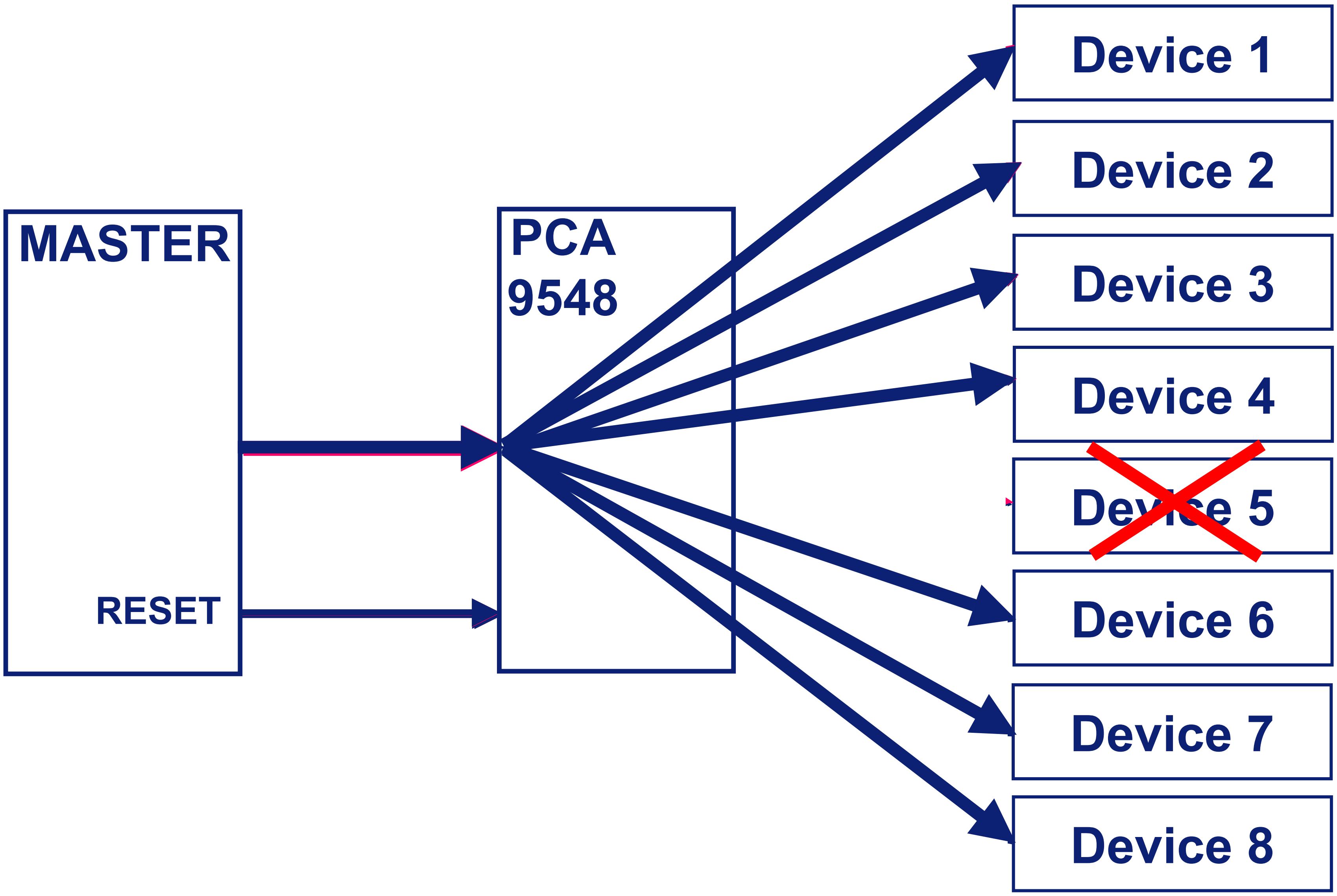
How to increase reliability of an I^{2}C bus? (Slave devices)

- I^{2}C protocol: If one device does not work properly and hangs the bus, then no device can be addressed anymore until the rogue device is separated from the bus or reset.

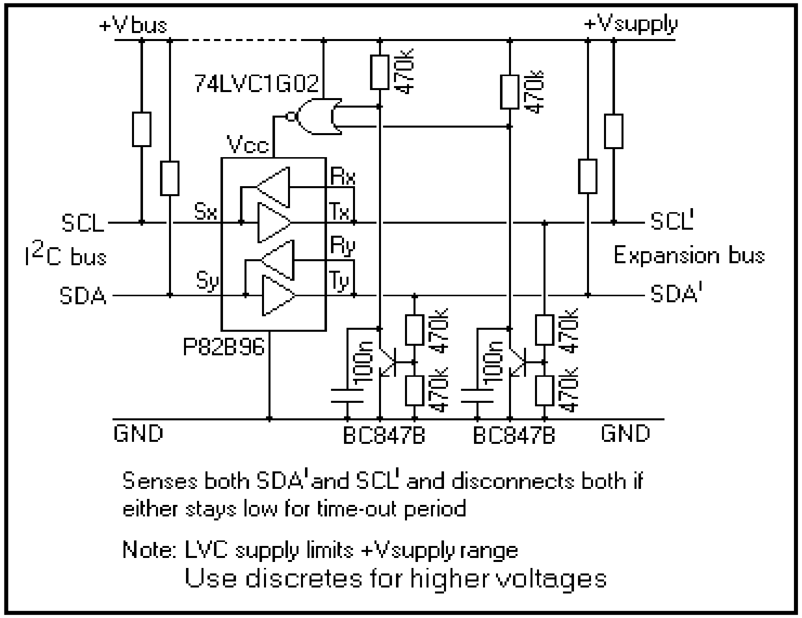
➔ **An I^{2}C switch can be used to split the I^{2}C bus in several branches that can be isolated if the bus hangs up.**

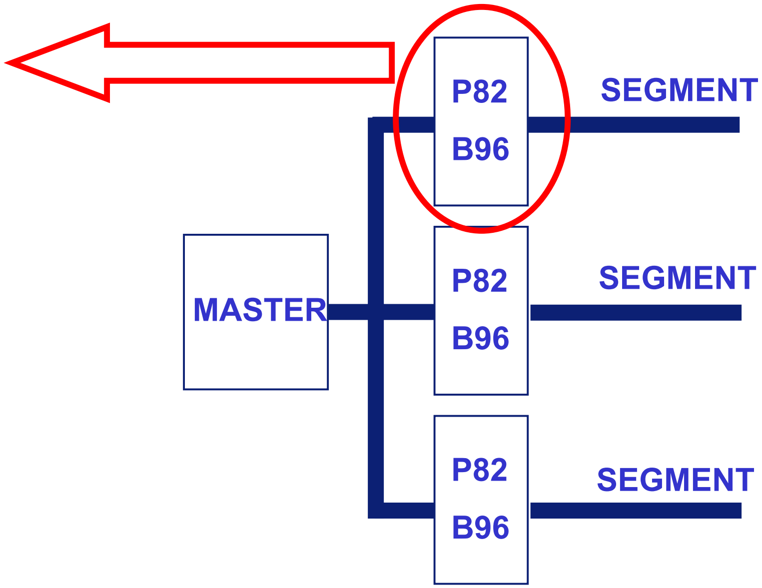
- Switches allow the main I^{2}C to be split dynamically in several sub-branches that can be:
  - active all the time
  - deactivated if one device of a particular branch hangs the bus
- When a malfunctioning sub-branch has been isolated, the other sub branches are still available
- It is programmable through I^{2}C so no additional pin is required to control it
- More than one switch can be plugged in the same I^{2}C bus

Isolate I²C hanging segment(s)



Isolate hanging segments
Discrete stand alone solution





- A bus buffer isolates the branch (capacitive isolation)
- Its power supply is controlled by a bus sensor
- SDA and SCL are sensed and the sensor generates a timeout when the bus stays low
- Bus buffer is Hi-Z when power supply is off.

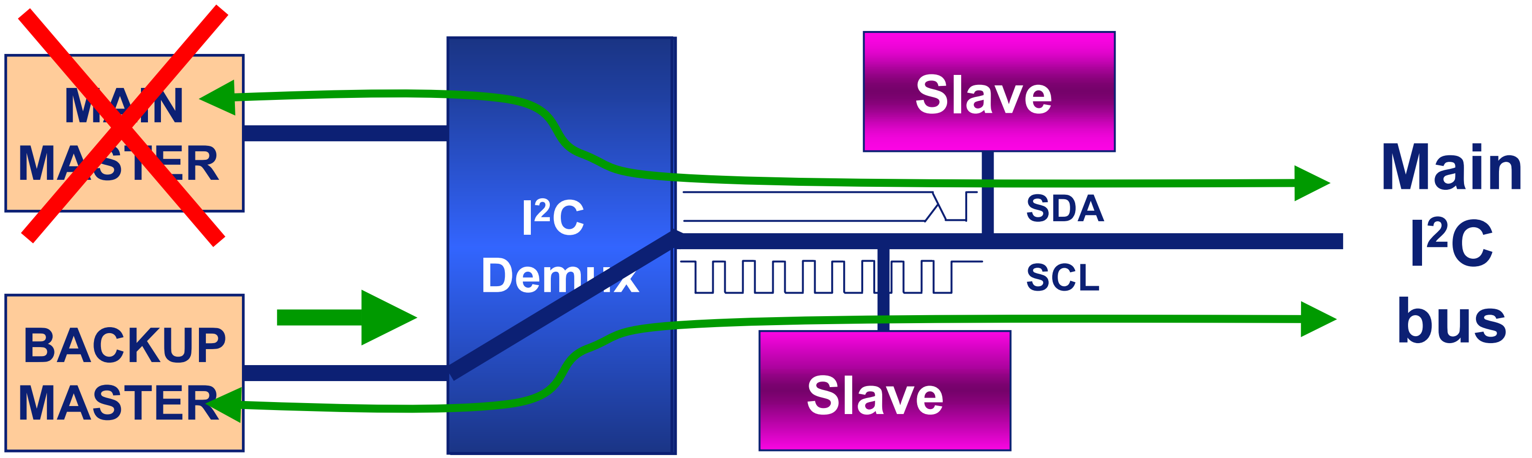
How to increase reliability of an I^2C bus? (Master devices)

- I^2C protocol: If the master does not work properly , reliability of the systems will decrease since monitoring or control of critical parameters are not possible anymore (voltage, temperature, cooling system)

➔ **An I^2C demultiplexer can be used to switch from one failing master to its backup.**

- It allows to have 2 independent masters to control the bus without any fault or system corruption
  - failed master completely isolated from the bus
  - I^2C bus is initialized by the demultiplexer before switching from one master to the other one
- It is programmable through I^2C so no additional pin is required to control it
- More than one demultiplexer can be plugged in the same I^2C bus

Isolate failing master



- Main Master control the I²C bus
- When it fails, backup master asks to take control of the bus
- Previous master is then isolated by the multiplexer
- Downstream bus is initialized (all devices waiting for START condition)
- Switch to the new master is done
- Products

Device	# of upstream channels
PCA9541	2

How to go beyond I²C max cap load?

• I²C protocol limitation: the maximum capacitive load in a bus is 400 pF. If the load is higher AC parameters will be violated.

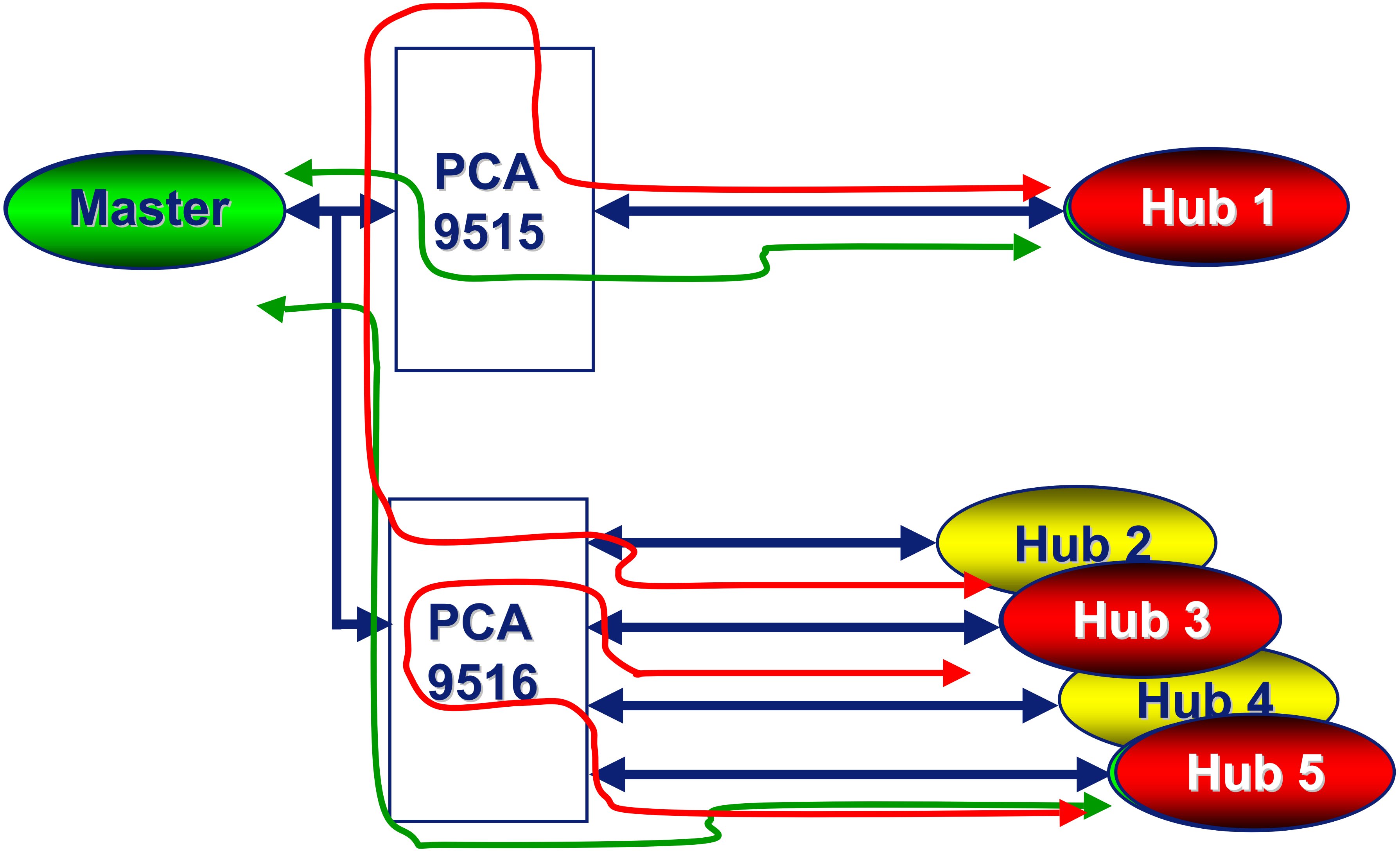
➔ **An I²C bus repeater or an I²C hub can be used to get rid of this limitation**

• It allows to double the I²C max capacitive load (repeater) or to make it 5 times higher (hub = 5 repeaters)
• Multi-master capable, voltage level translation
• All channels can be active at the same time
• Limitation: Repeater/hub cannot be used in series

• Products:

Device	# of repeaters	# of ENABLE pins
PCA9515	1	1
PC9516	5	4

I²C Bus repeater (PCA9515) and Hub (PCA9516)



How to scale the I^2C bus by adding 400 pF segments?

- Some applications require architecture enhancements where one or several isolated I^2C hubs need to be added with the capability of hub to hub communication

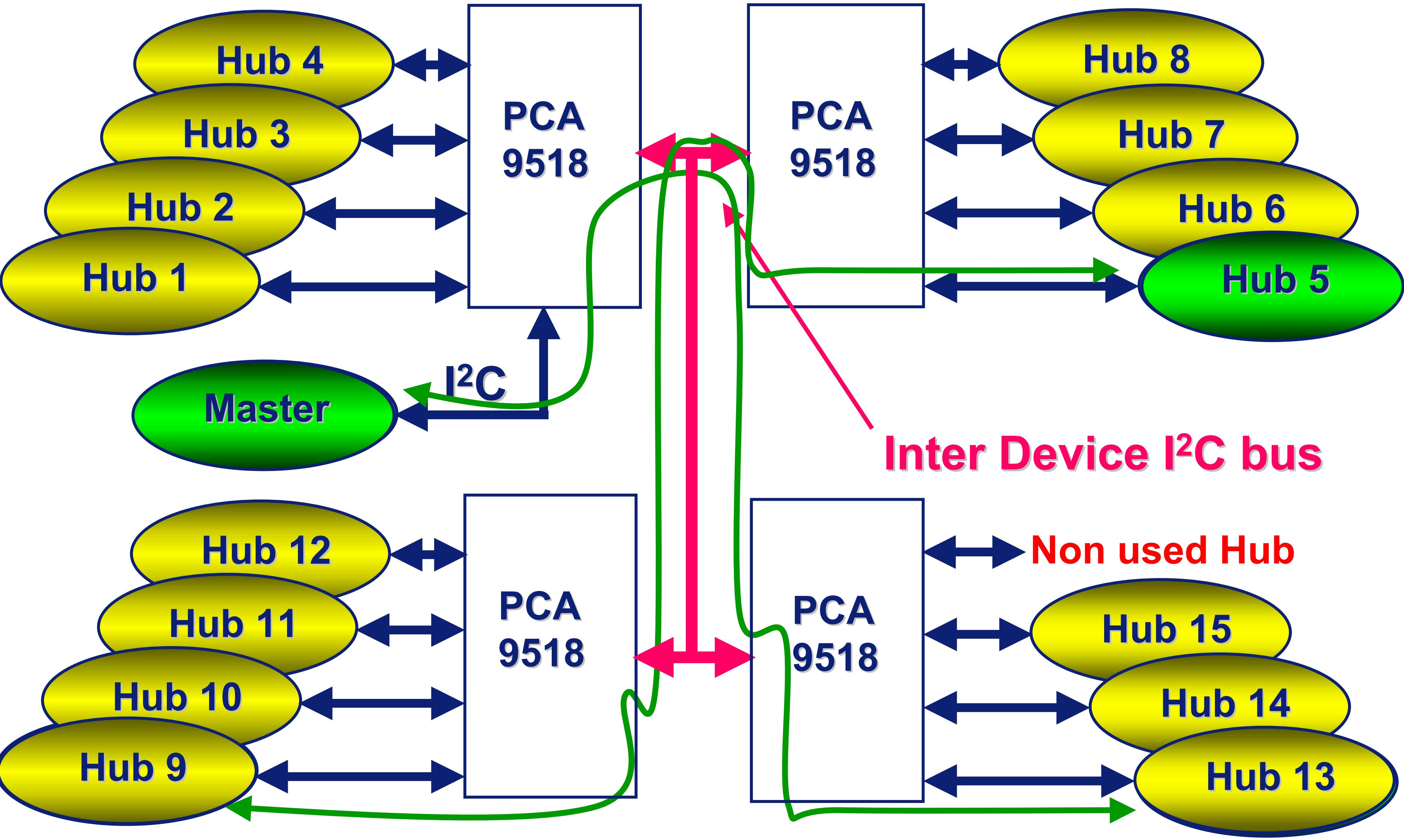
➔ **An expandable I^2C hub can be used to easily upgrade this type of application**

- It allows to expand the numbers of hubs without any limit
- Multi-master capable, voltage level translation
- All channels can be active at the same time (4 channels per expandable hub can be individually disabled)

- Products:

Device	# of repeaters	# of ENABLE pins
PCA9518	5	4

PCA9518 Applications



How to accommodate 100 kHz and 400 kHz devices in the same I^2C bus?

- I^2C protocol limitation: in an application where 100 kHz and 400 kHz devices (masters and/or slaves) are present in the same bus, the lowest frequency must be used to guarantee a safe behavior.

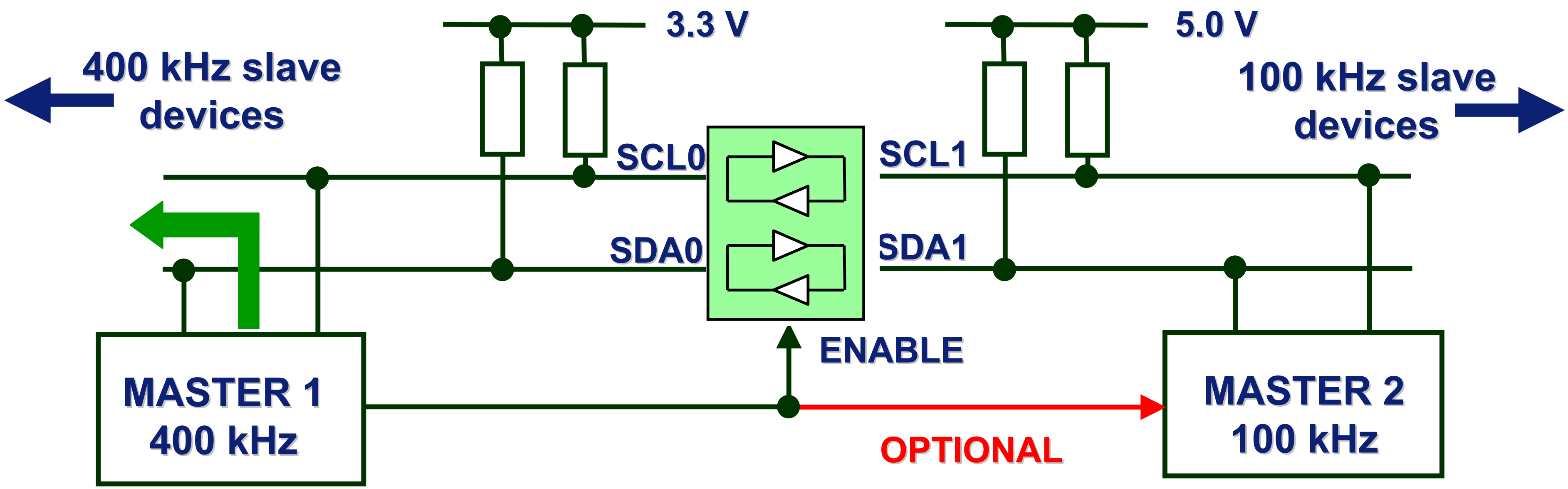
➔ **An I^2C bus repeater can be used to isolate 100 kHz from 400 kHz devices when a 400 kHz communication is required**

- It allows to easily upgrade applications where legacy 100 kHz I^2C devices share bus access with newer 400 kHz I^2C devices
- Each side of the repeater can work with different logic voltage levels

- Products:

Device	# of repeaters	# of ENABLE pins
PCA9515	1	1

PCA9515 - Application Example



- Master 1 works at 400 kHz and can access 100 & 400 kHz slaves at their maximum speed (100 kHz only for 100 kHz devices)
- Master 2 works at only 100 kHz
- PCA9515 is disabled (ENABLE = 0) when Master 1 sends commands at 400 kHz

How to live insert?

• I^2C protocol limitation: in an application where the I^2C bus is active, it was not designed for insertion of new devices.

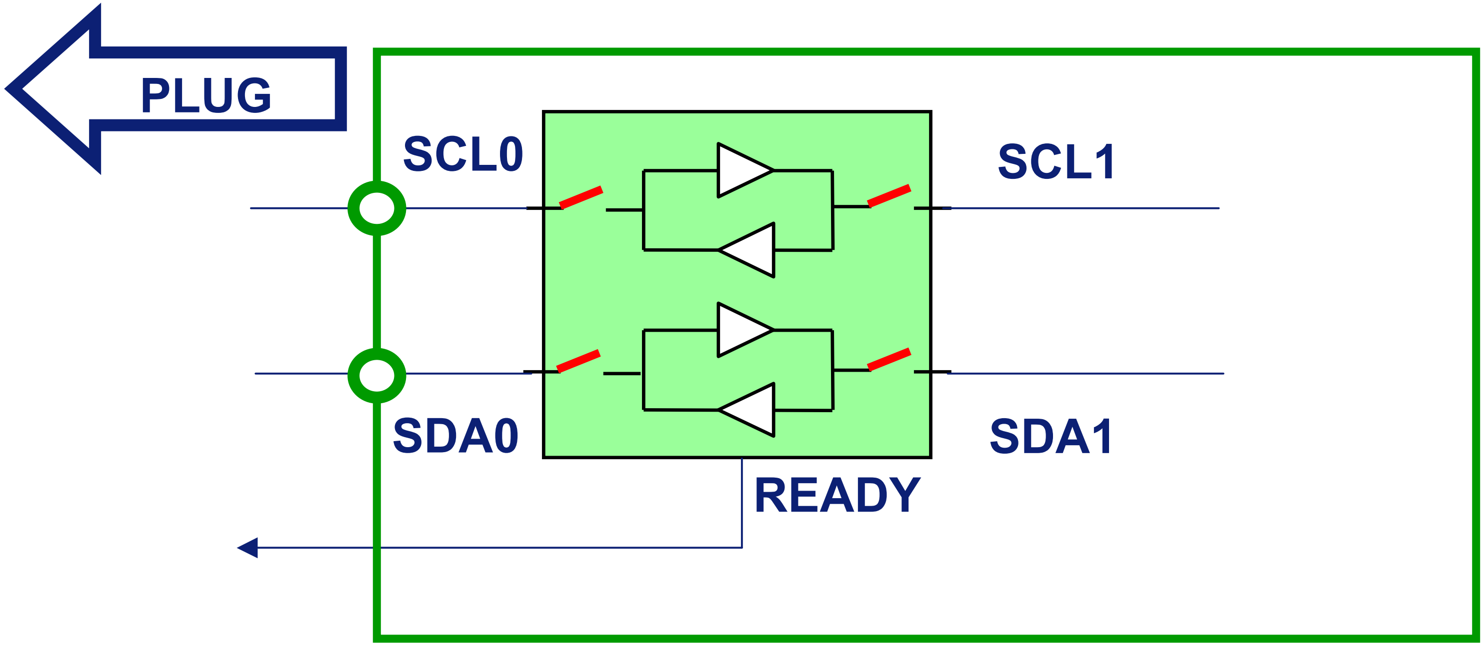
➔ An I^2C hot swap bus buffer can be used to detect bus idle condition isolate capacitance, and prevent glitching SDA & SCL when inserting new cards into an active backplane.

• Repeaters work with the same logic level on each side except the PCA9512 which works with 3.3 V and 5 V logic voltage levels at the same time

• Products:

Device	# of repeaters	# of ENABLE pins
PCA9511	1	1
PCA9512	1	0
PCA9513	1	1
PCA9514	1	1

I²C Hot Swap Bus Buffer



- Card is plugged on the system - Buffer is on Hi-Z state
  - Bus buffer checks the activity on the main I²C bus
    - When the bus is idle, upstream and downstream buses are connected
      - Ready signal informs that both buses are connected together

How to send I²C commands through long cables?

- I²C limitation: due to the bus 400 pF maximum capacitive load limit, sending commands over wire (80 pF/m) long distances is hard to achieve

➔ An I²C bus extender can be used

- It has high drive outputs
- Possible distances range from 50 meters at 85 kHz to 1km at 31 kHz over twisted-pair phone cables. Up to 400 kHz over short distances.

- Others applications:
  - Multi-point applications: link applications, factory applications
  - I²C opto-electrical isolation
  - Infra-red or radio links

- Products:

Device
P82B715
P82B96

How to use a micro-controller without I^2C bus or how to develop a dual master application with a single micro-controller?

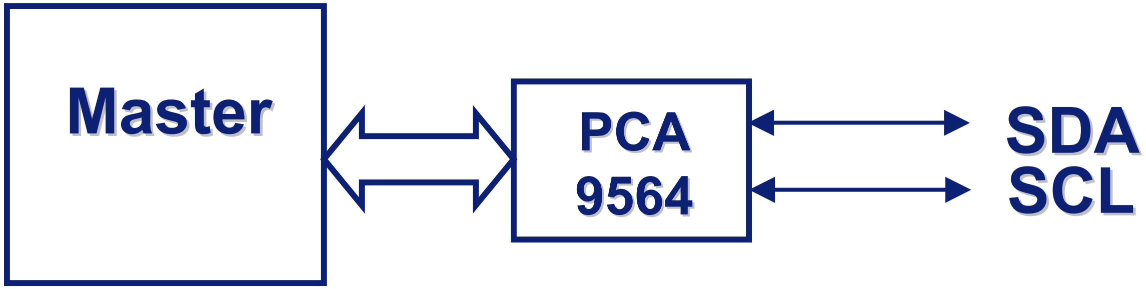
- Some micro-controllers integrates an I^2C port, others don't

➔ An I^2C bus controller can be used to interface with the micro-controller's parallel port

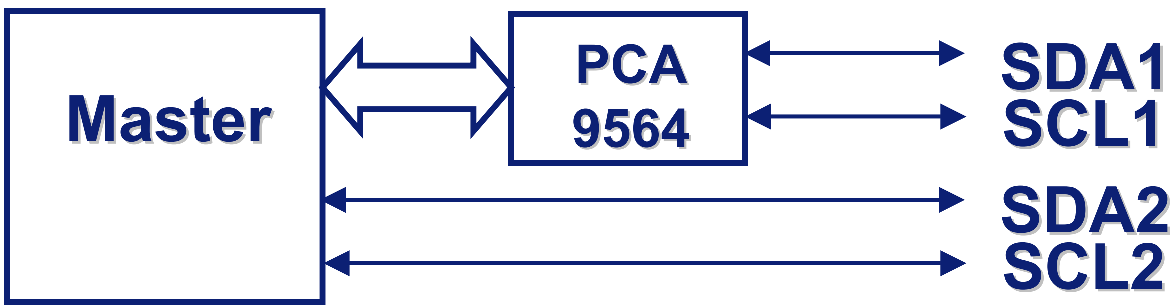
- It generates the I^2C commands with the instructions from the micro controller's parallel port (8-bits)
- It receives the I^2C data from the bus and send them to the micro-controller
- It converts by software any device with a parallel port to an I^2C device

Parallel Bus to I²C Bus Controller

- Master without I²C interface



- Multi-Master capability or 2 isolated I²C bus with the same device



- Products

	Voltage range	Max I²C freq	Clock source	Parallel interface
PCF8584	4.5 - 5.5V	90 kHz	External	Slow
PCA9564	2.3 - 3.6V w/5V tolerance	360 kHz	Internal	Fast

Development Tools and Evaluation Board Overview

Purpose of the Development Tool and I^2C Evaluation Board

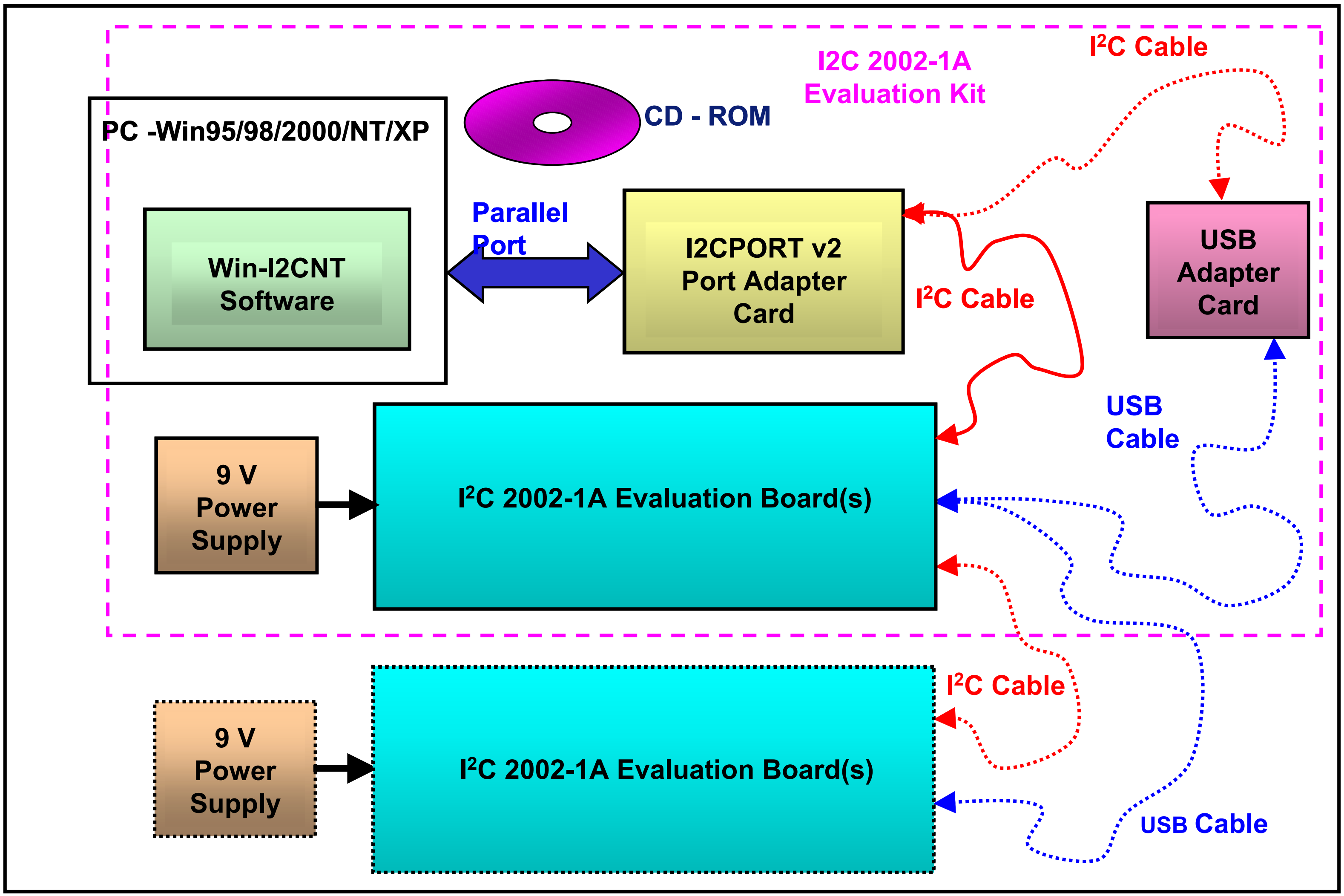
To provide a low cost platform that allows Field Application Engineers, designers and educators to easily test and demonstrate I^2C devices in a platform that allows multiple operations to be performed in a setting similar to a real system environment.

I2C 2002-1A Evaluation Board Kit

FEATURES

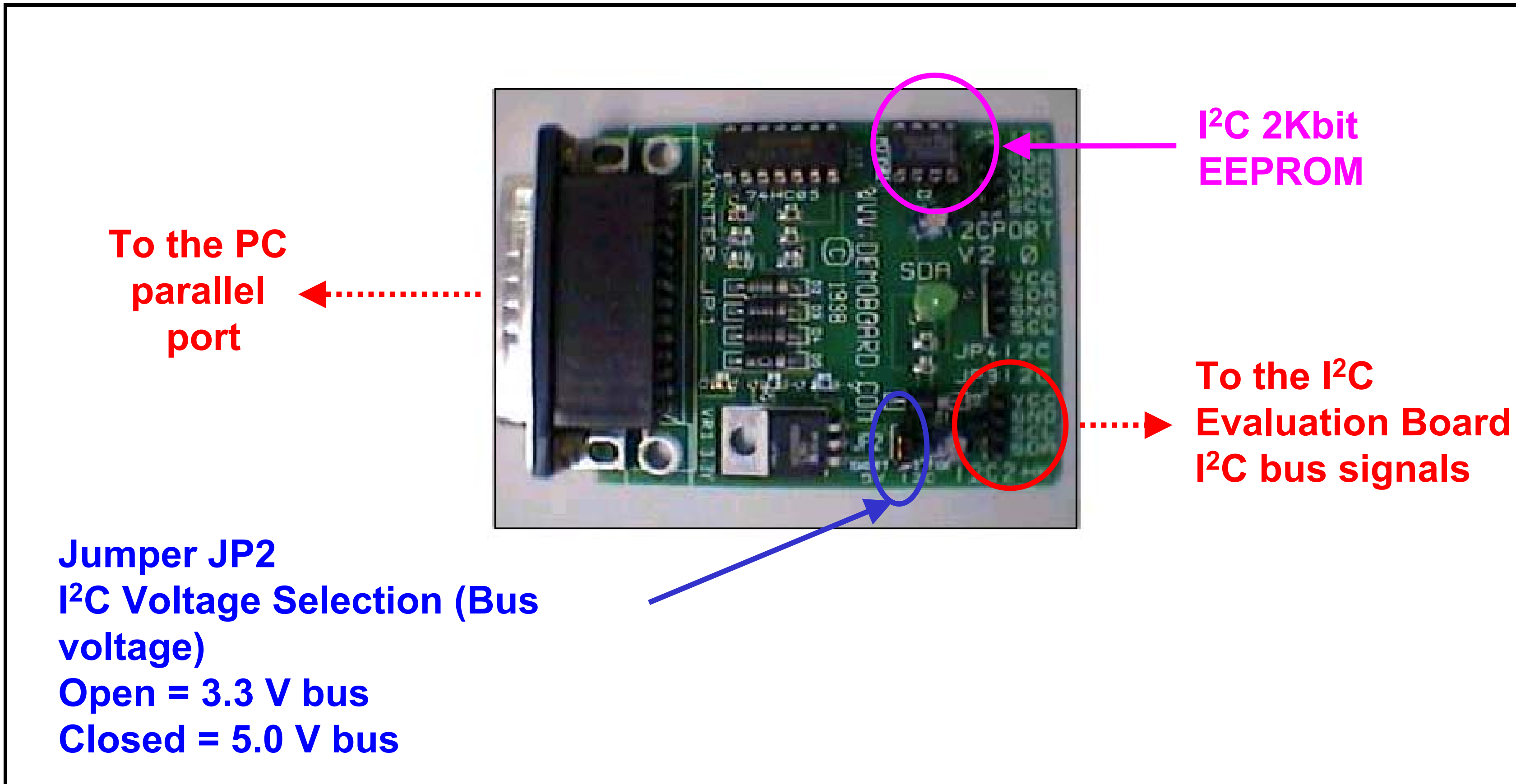
- Converts Personal Computer parallel port to I²C bus master
- Simple to use graphical interface for I²C commands
- Win-I2CNT software compatible with Windows 95, 98, ME, NT, XP and 2000
- Order kits at www.demoboard.com

Evaluation Board 2002-1A Kit Overview

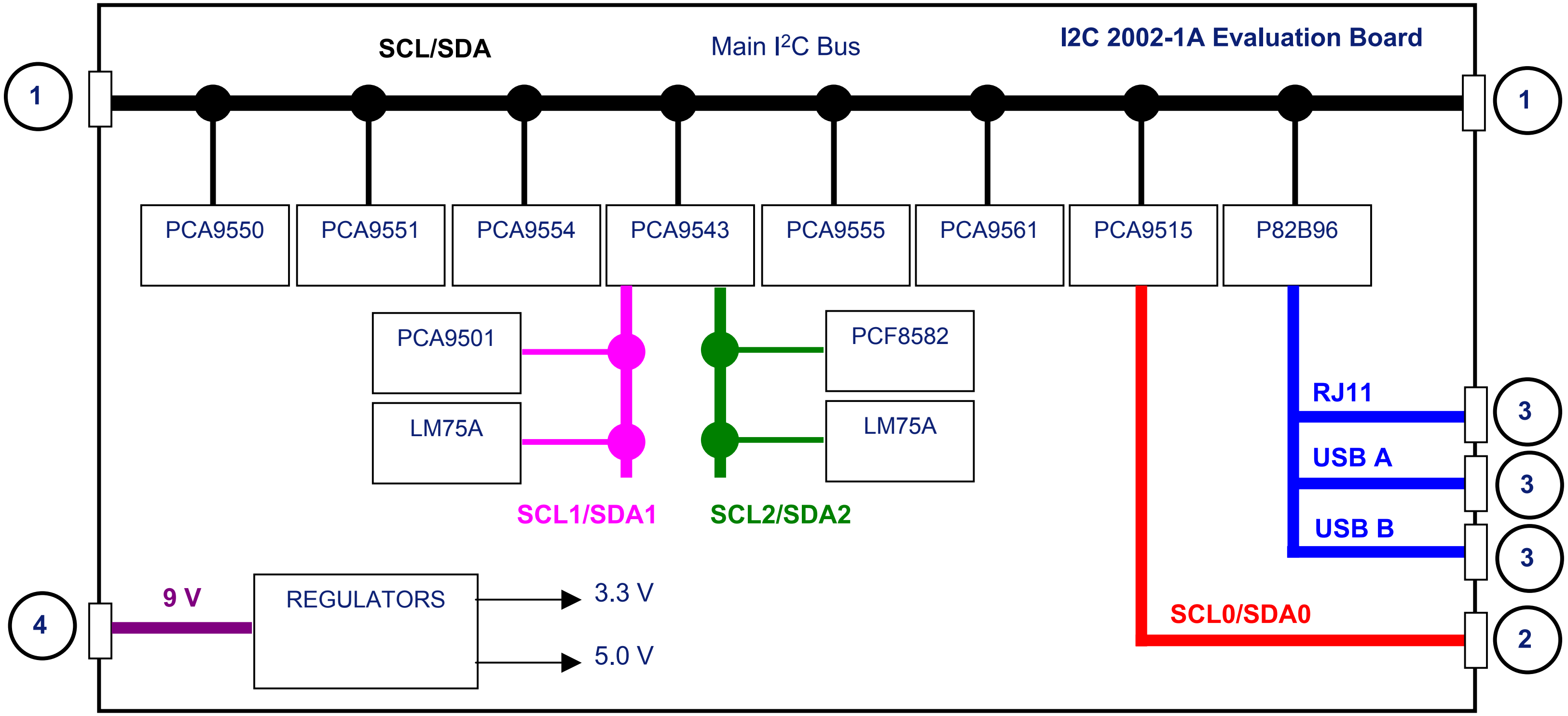


I2CPORT v2 Adapter Card

- **The Win-I2CNT adapter connects to the standard DB-25 on any PC**
- **It can be powered by the PC or by the evaluation board**



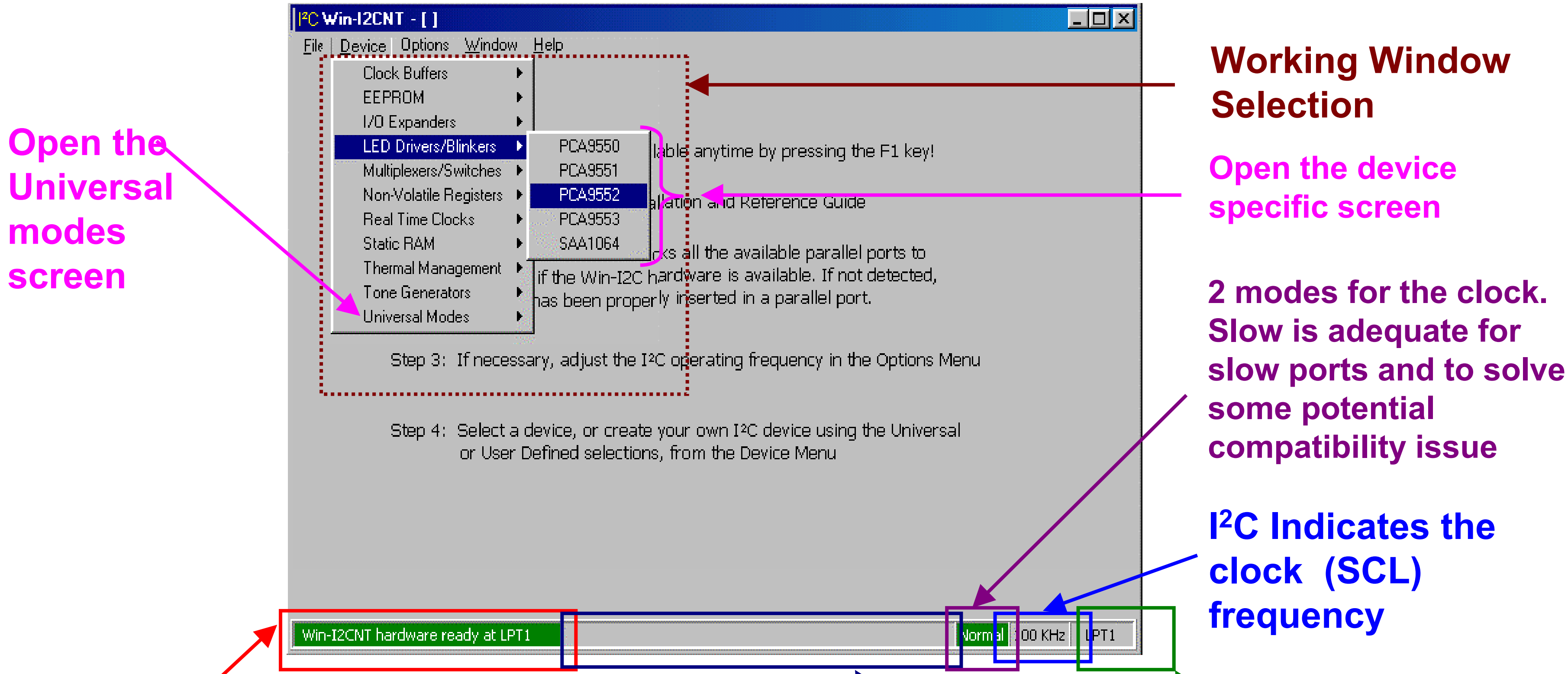
Evaluation Board I2C 2002-1A Overview



- 12 I^{2}C devices on the evaluation board
- 2 evaluation boards can be daisy chained without any address conflict
- Boards cascadable through I^{2}C connectors, RJ11 phone cable or USB cable
- On board regulators

Starting the Software

Clicking on the Win- I2CNT icon will start the software and will give the following window



Indicates that I²C communications can start
If problem, message "WIN-I2C hardware not detected" displayed
→ Action: check Adapter Card

Device → I/O Expanders → PCA9501

GPIO register value

GPIO value

GPIO Read / Write Options

GPIO programming

GPIO address

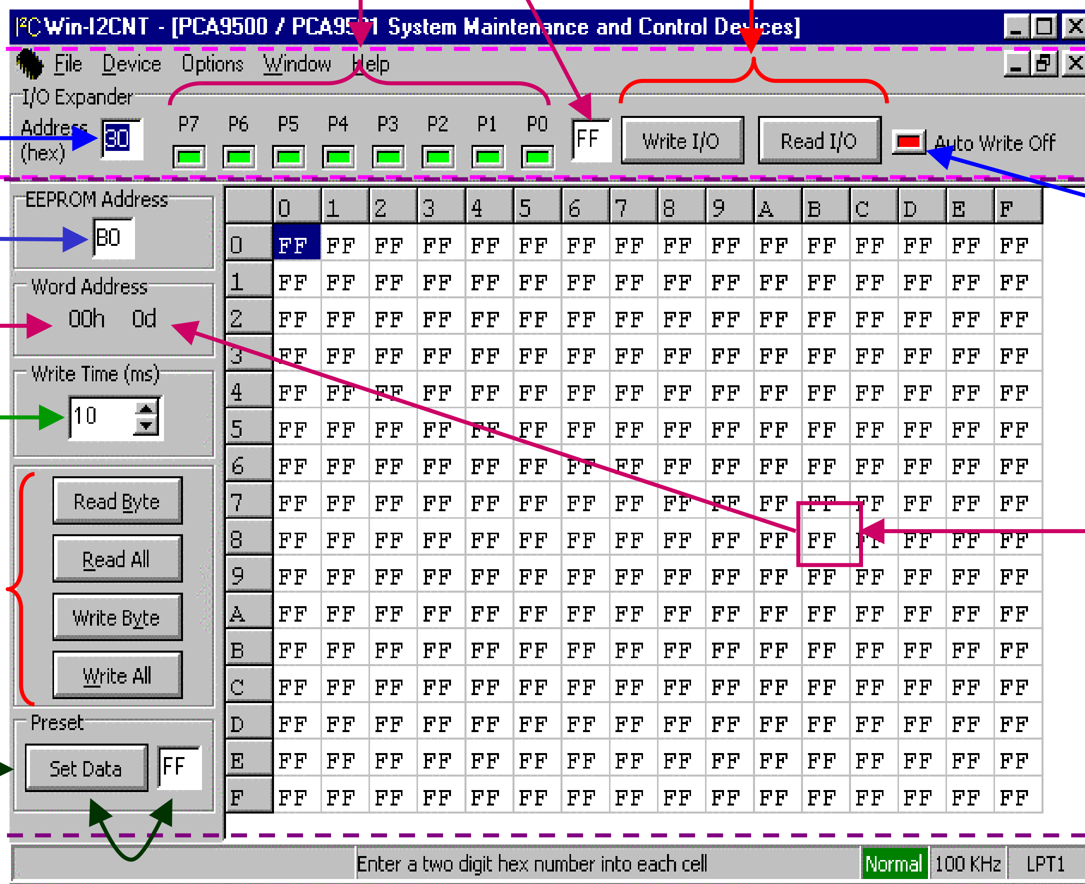
EEPROM address

Selected byte information

Write Time

EEPROM Read / Write Options

Set the all EEPROM to the same value

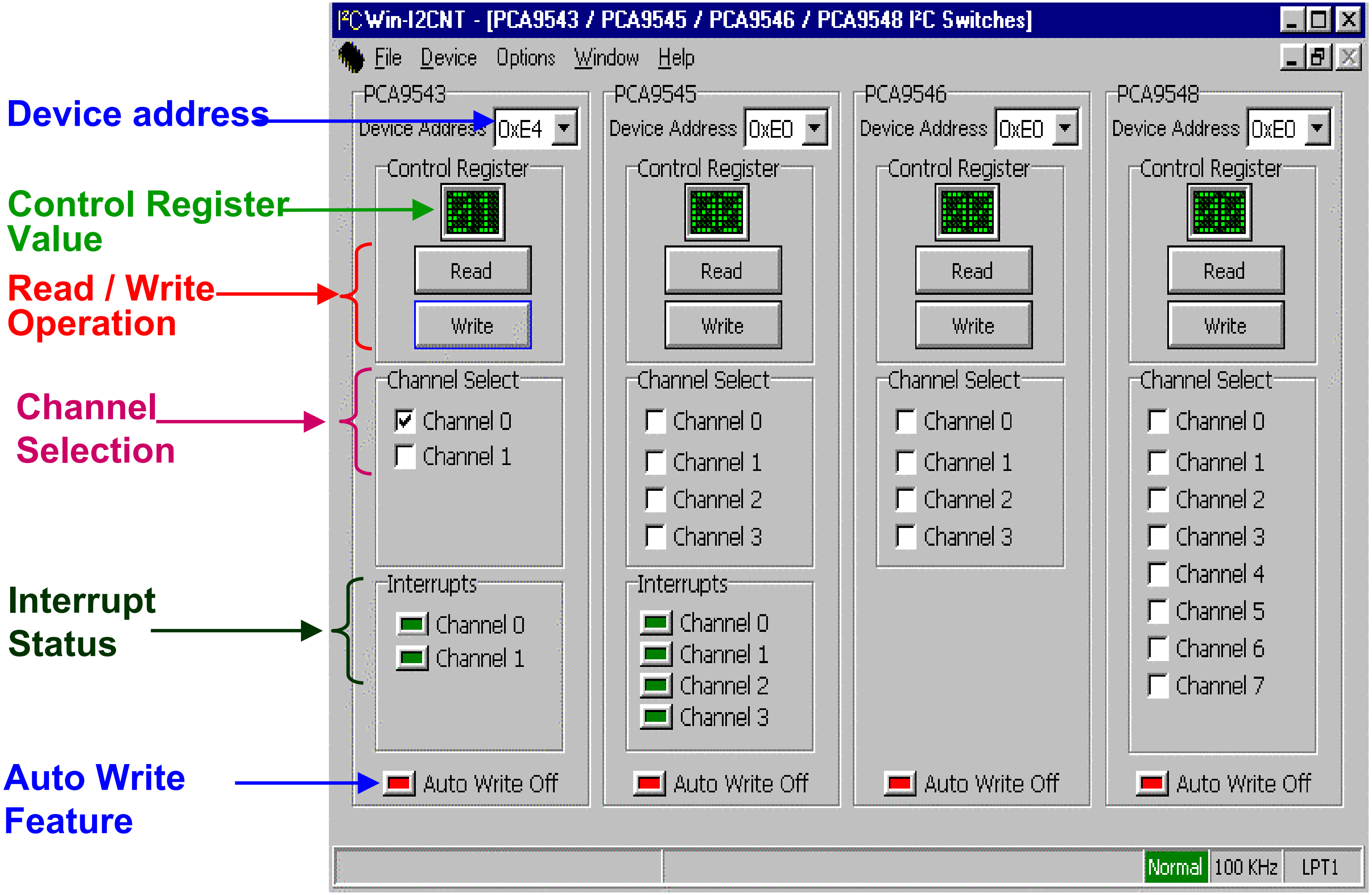


Auto Write Feature

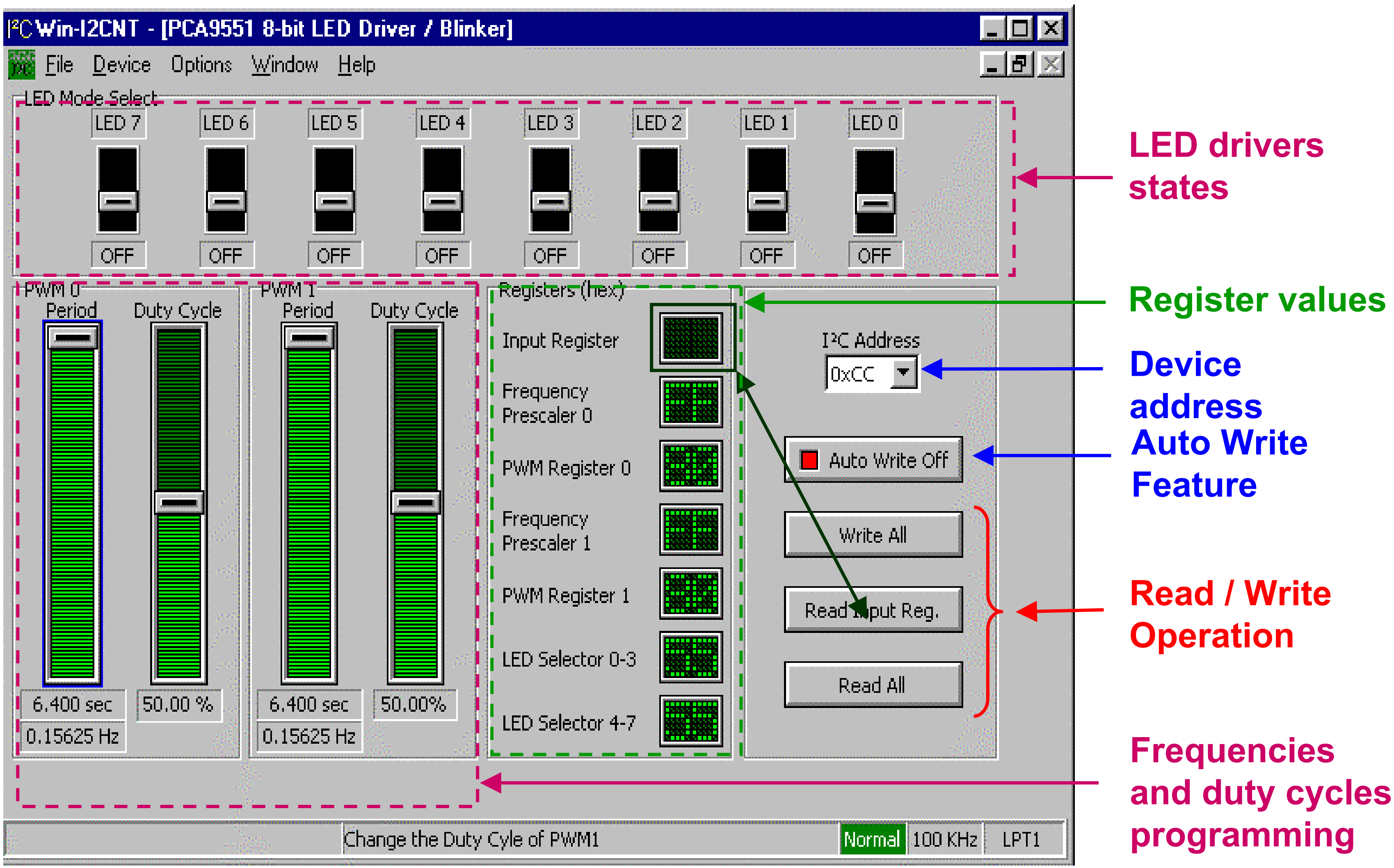
Byte $8B_H$ or 139_{10}

EEPROM programming

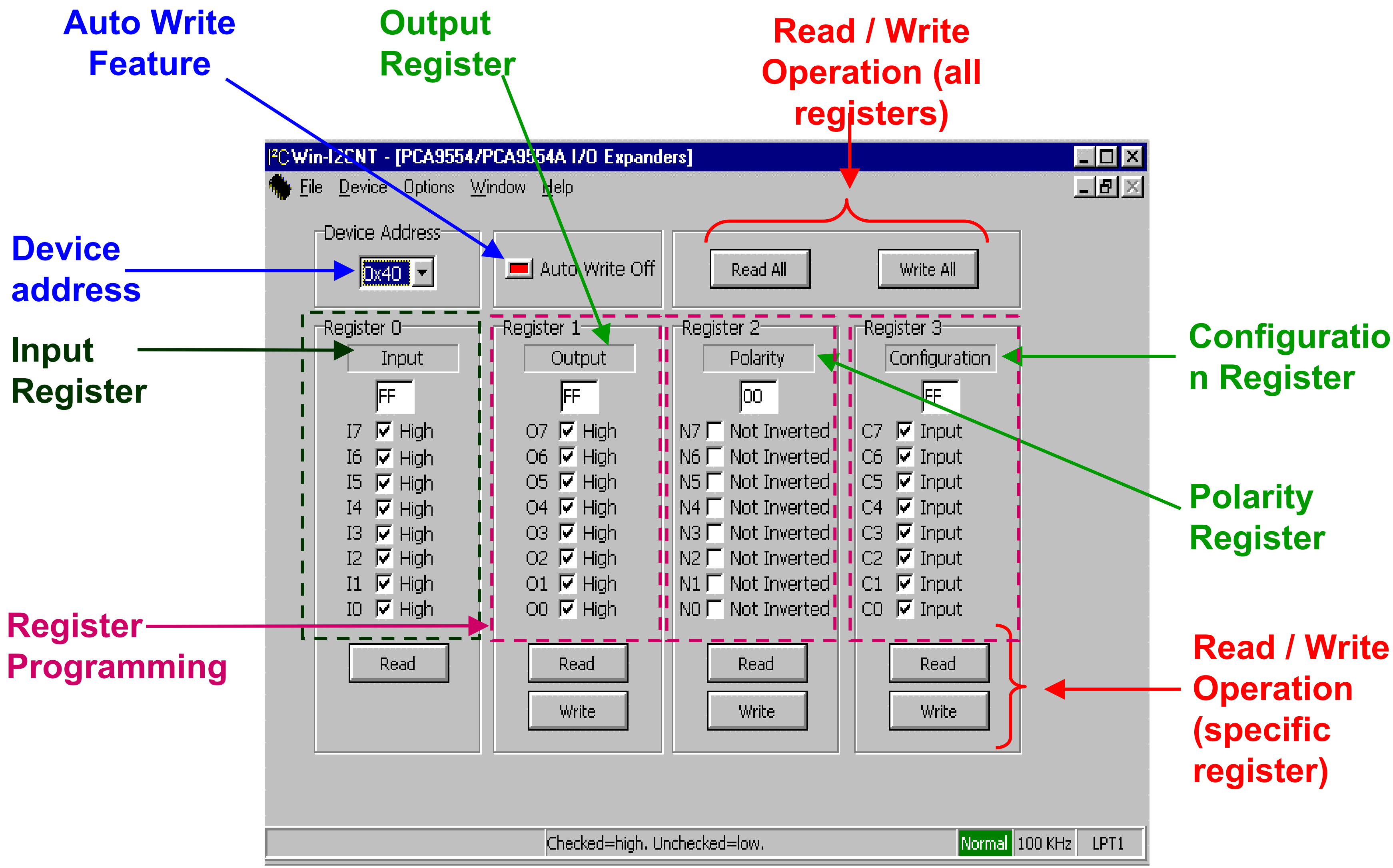
Device → Multiplexers/Switches → PCA9543



Device → LED Drivers/Blinkers → PCA9551



Device → I/O Expanders → PCA9554



Device → I/O Expanders → PCA9555

Auto Write Feature

Polarity Registers

Input Registers

Read / Write Operation (all registers)

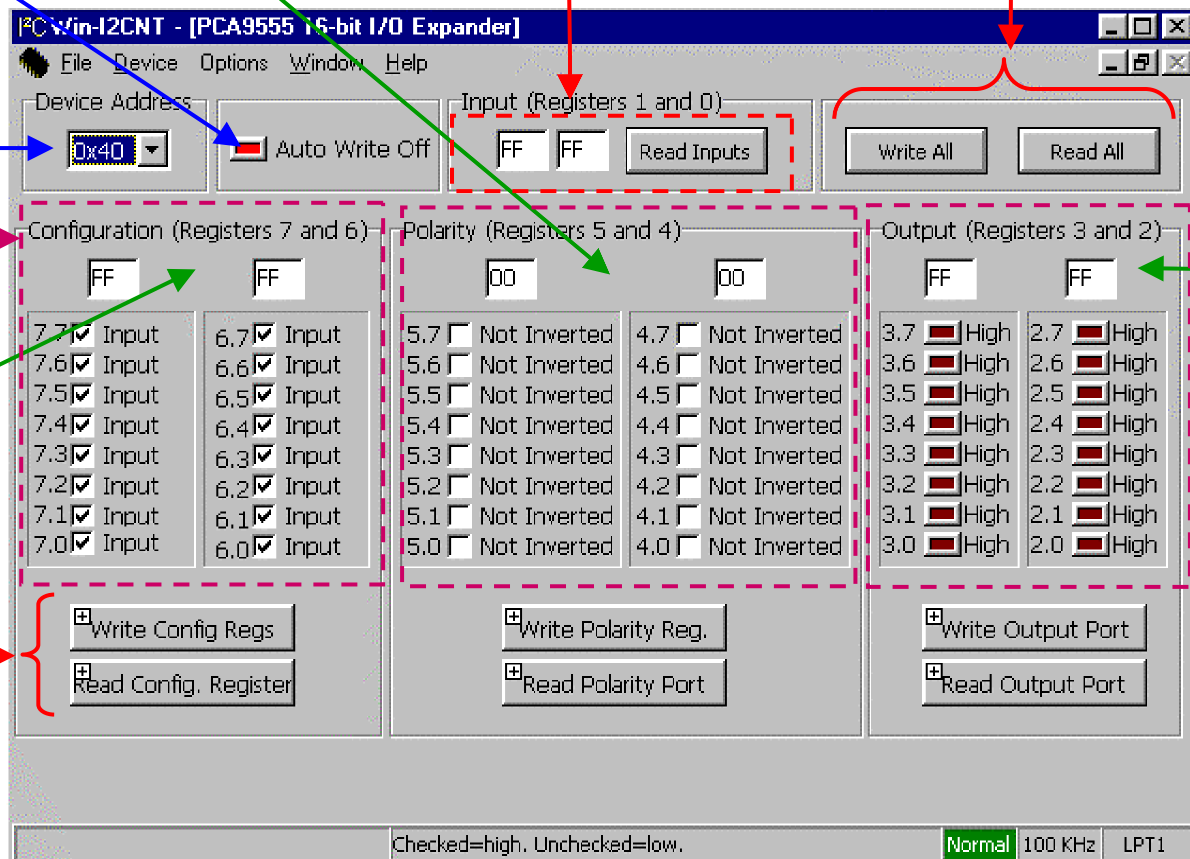
Device Address

Register Programming

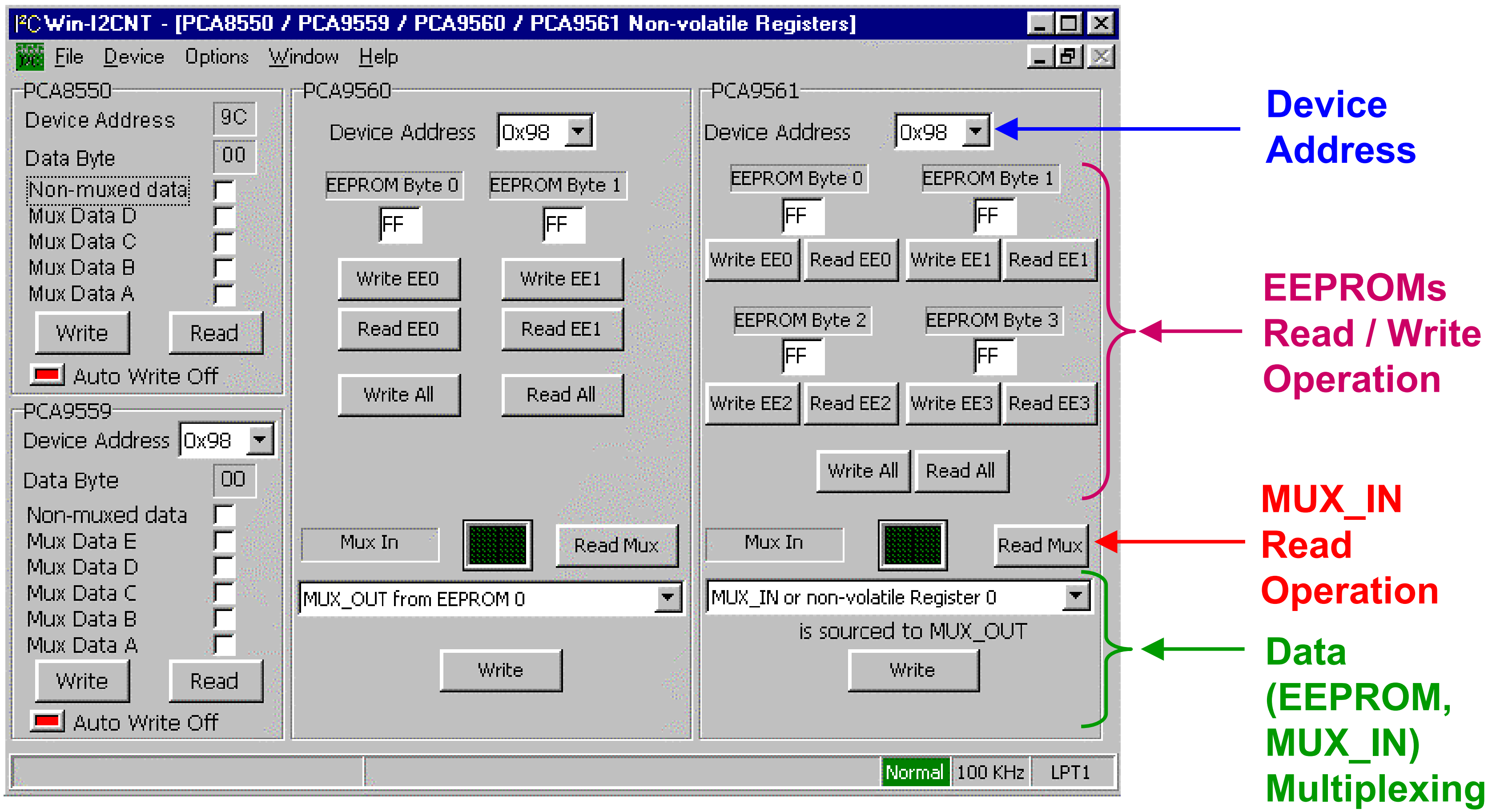
Configuration Registers

Read / Write Operation (specific Register)

Output Registers

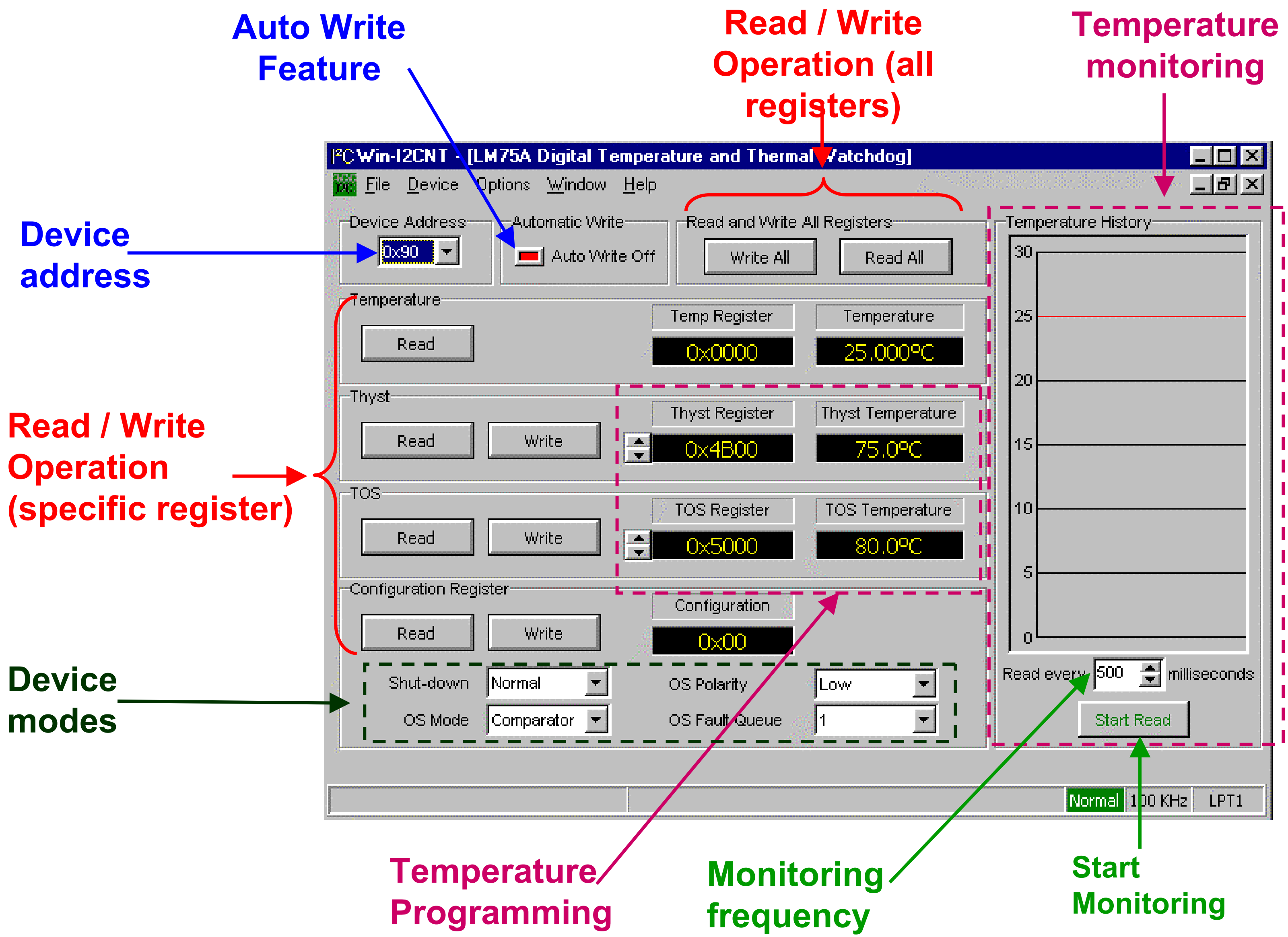


Device → Non-Volatile Registers → PCA9561



Note: MUX_IN, MUX_SELECT and WP pins are not controlled by the Software

Device → Thermal Management → LM75A



Device → EEPROM → 256 x 8 (2K)

- **Control window and operating scheme same as PCA9501's 2KBit EEPROM**

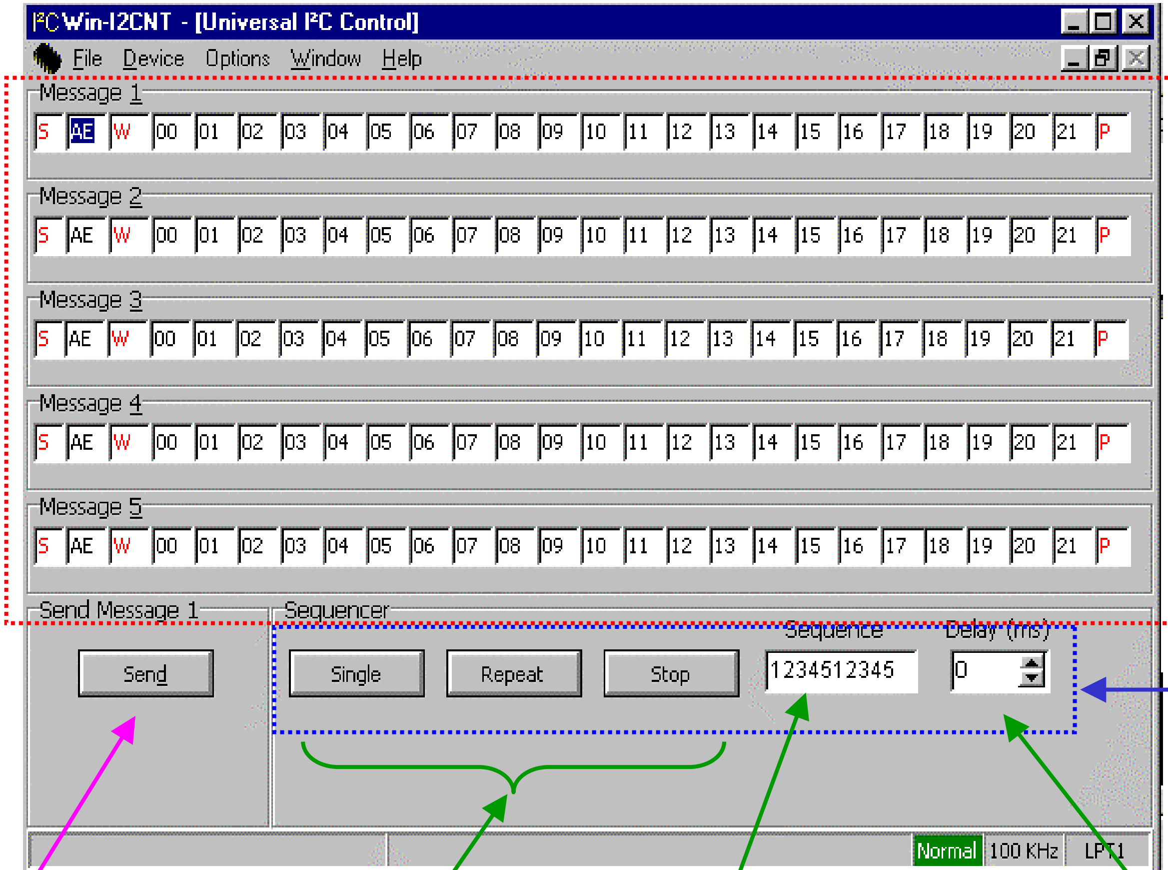
PCA9515

- **Bus repeater - No software to control it**
- **Buffered I²C connector available**
- **Enable Control pin accessible**

P82B96

- **Bus buffer - No software to control it**
- **I²C can come from the Port Adapter + USB Adapter through the USB cable**
- **I²C can be sent through RJ11 and USB cables to others boards**
- **5.0 V and 9.0 V power supplies**

Universal Receiver / Transmitter Screen



Commands Programming

I²C sequencing parameters

Send selected message

Sequencer

Sequence programming

Programmable delay between the messages

How to program the Universal Screen?

- Length of the messages is variable: 20 instructions max
- 5 different messages can be programmed
- First START and STOP instructions can not be removed
- I^2C Re-Start Command → "S" key
- I^2C Write Command → "W" key
- I^2C Read Command → "R" key
- Add an Instruction → "Insert" key
- Remove an Instruction → "Delete" key
- Data: 0 to 9 + A to F keys

Some others interesting Features

- I^2C clock frequency can be modified (Options Menu).
- Acknowledge can be ignored for stand alone experiment (Options Menu).
- Universal Transmitter/Receiver program can be saved in a file.
- Device specific screens are different depending on the selected device. All the options are usually covered in those screens. Good tool to learn how the devices work and test all the features.
- Possibility to build some small applications by connecting the devices together through the headers.

How To Obtain the New Evaluation Kit

- The I2C 2002-1A Evaluation Board Kit consists of the:
  - I2C 2002-1A Evaluation Board
  - I2CPort v2 Adapter Card for the PC parallel port
  - 4-wire connector cable
  - USB Adapter Card (no USB cable included)
  - 9 V power supply
  - CD-ROM with operating instructions and Win-I2CNT software on that provides easy to use PC graphical interface specific to the I^2C devices on the evaluation board but also with general purpose mode for all other I^2C devices.

Purchase the I2C 2002-1A Evaluation Board Kit at www.demoboard.com

3^{rd} Hour

Comparison of I^2C with SMBus

Some words on SMBus

- Protocol derived from the I^2C bus
- Original purpose: define the communication link between:
  - an intelligent battery
  - a charger
  - a microcontroller
- Most recent specification: Version 2.0
  - Include a low power version and a “normal” power version
  - can be found at: **www.SMBus.org**
- Some minor differences between I^2C and SMBus:
  - Electrical
  - Timing
  - Operating modes

I^2C Bus Vs SMBus - Electrical Differences

DC parameter comparison between Standard I^2C, Fast I^2C and SMBus devices								
Symbol	Parameter	Std I^2C mode device		Fast I^2C mode device		SMBus device		Units
		MIN	MAX	MIN	MAX	MIN	MAX	
V_{IL}	Fixed input level	-0.5	1.5	-0.5	1.5	-	0.8	V
	V_{DD} related input level	-0.5	$0.3V_{DD}$	-0.5	$0.3\ V_{DD}$	N/A	N/A	V
V_{IH}	Fixed input level	3.0	V_{DD}max+ 0.5	3.0	V_{DD}max+0 .5	2.1	5.5	V
	V_{DD} related input level	$0.7V_{DD}$	V_{DD}max+ 0.5	$0.7V_{DD}$	V_{DD}max+0 .5	N/A	N/A	V
V_{HYS}	V_{IH}-V_{IL}	N/A	N/A	$0.05V_{DD}$	-	N/A	N/A	V
V_{OL}	V_{OL} @ 3mA	0	0.4	0	0.4	N/A	N/A	V
	V_{OL} @ 6mA	N/A	N/A	0	0.6	N/A	N/A	
	V_{OL} @ 350uA	N/A	N/A	N/A	N/A	-	0.4	
I_{PULLUP}		N/A	N/A	N/A	N/A	100	350	uA
I_{LEAK}		-10	10	-10	10	-5	5	uA

Low Power version of the SMBus Specification only

The SMBus specification can be found on SMBus web site at www.SMBus.org

I^2C Bus Vs SMBus - Timing and operating modes Differences

- Timing:
  - Minimum clock frequency = 10 kHz
  - Maximum clock frequency = 100 kHz
  - Clock timeout = 35 ms
- Operating modes
  - slaves must acknowledge their address all the time (mechanism to detect a removable device's presence)

Intelligent Platform Management Interface (IPMI)

Intelligent Platform Management Interface

- Intel initiative in conjunction with hp, NEC and Dell
- Initiative consists of three specifications:
  - IPMI for software extensions
  - Intelligent Platform Management Bus (IPMB) for intra-chassis (in side the box) extensions
  - Inter Chassis Management Bus (ICMB) for inter-chassis (outside of the box) extensions
- Needed since as the complexity of systems increase, MTBF decreases
- Defines a standardized, abstracted, message-based interface to intelligent platform management hardware.
- Defines standardized records for describing platform management devices and their characteristics.
- Provides a self monitoring capability increasing reliability of the systems

Intelligent Platform Management Interface

- IPMI
  - Provides a self monitoring capability increasing reliability of the systems
  - Monitor server physical health characteristics :
    - temperatures
    - voltages
    - fans
    - chassis intrusion
  - General system management:
    - automatic alerting
    - automatic system shutdown and re-start
    - remote re-start
    - power control
- More information – www.intel.com/design/servers/ipmi/ipmi.htm

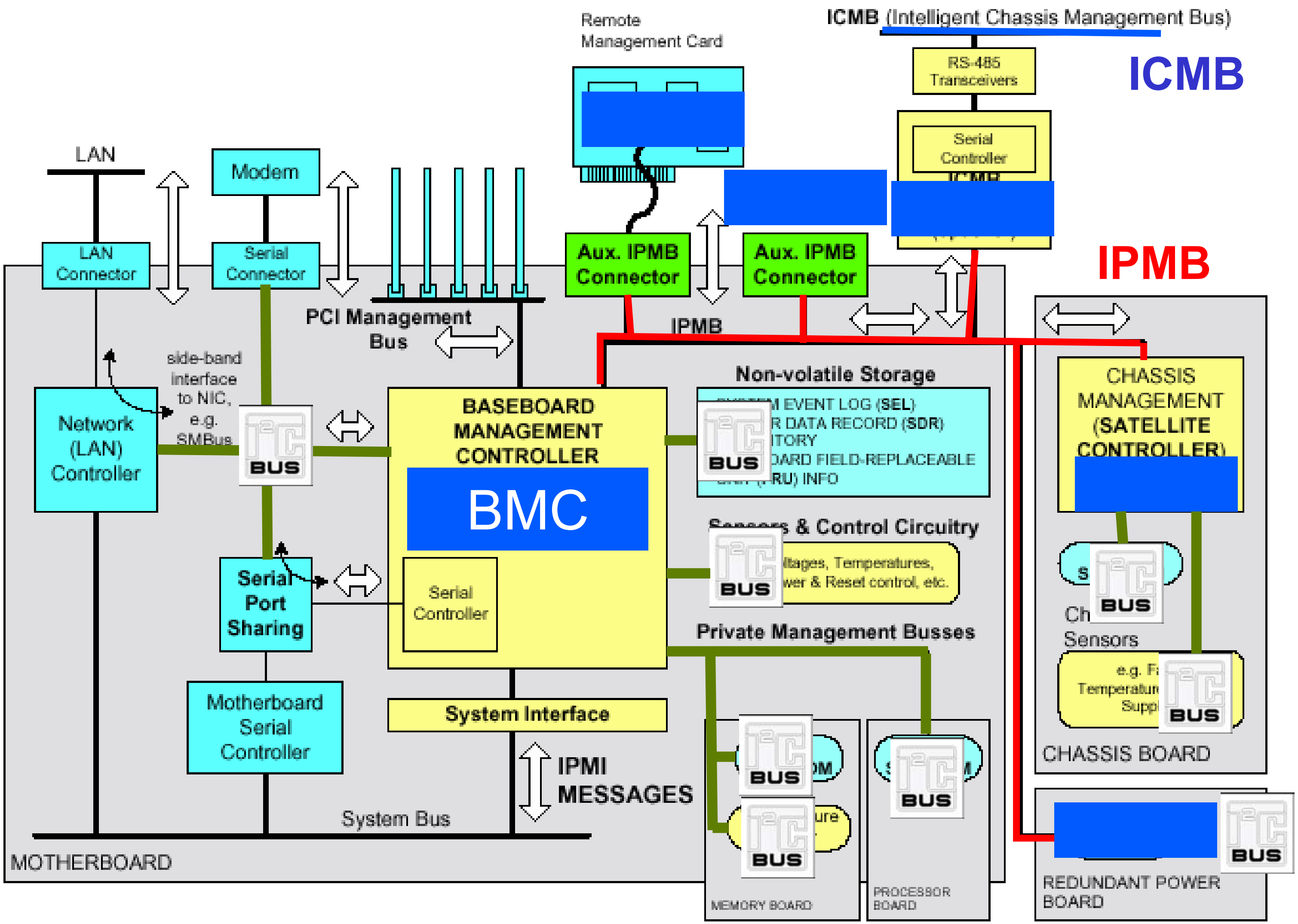
Intelligent Platform Management Bus

- Standardized bus and protocol for extending management control, monitoring, and event delivery **within** the chassis:
  - I^2C based
  - Multi-master
  - Simple Request/Response Protocol
  - Uses IPMI Command sets
  - Supports non-IPMI devices
- Physically I^2C but write only (master capable devices), hot swap not required.
- Enables the Baseboard Management Controller (BMC) to accept IPMI request messages from other management controllers in the system.
- Allows non-intelligent devices as well as management controllers on the bus.
- BMC serves as a controller to give system software access to IPMB

IPMI Details

- Defines a standardized interface to intelligent platform management hardware
  - Prediction and early monitoring of hardware failures
  - Diagnosis of hardware problems
  - Automatic recovery and restoration measures after failure
  - Permanent availability management
  - Facilitate management and recovery
  - Autonomous Management Functions: Monitoring, Event Logging, Platform Inventory, Remote Recovery
  - Implemented using Autonomous Management Hardware: designed for Microcontrollers based implementations
- Hardware implementation is isolated from software implementation
- New sensors and events can then be added without any software changes

Overall IPMI Architecture



Where IPMI is being used

Intel Server Management

Servers today run mission-critical applications. There is literally no time for downtime. That is why Intel created Intel® Server Management – a set of hardware and software technologies built right into most Intel® sever boards that monitors and diagnoses server health. Intel Server Management helps give you and your customers more server uptime, increased peace of mind, lower support costs, and new revenue opportunities.

More information:

program.intel.com/shared/products/servers/boards/server_management

PICMG

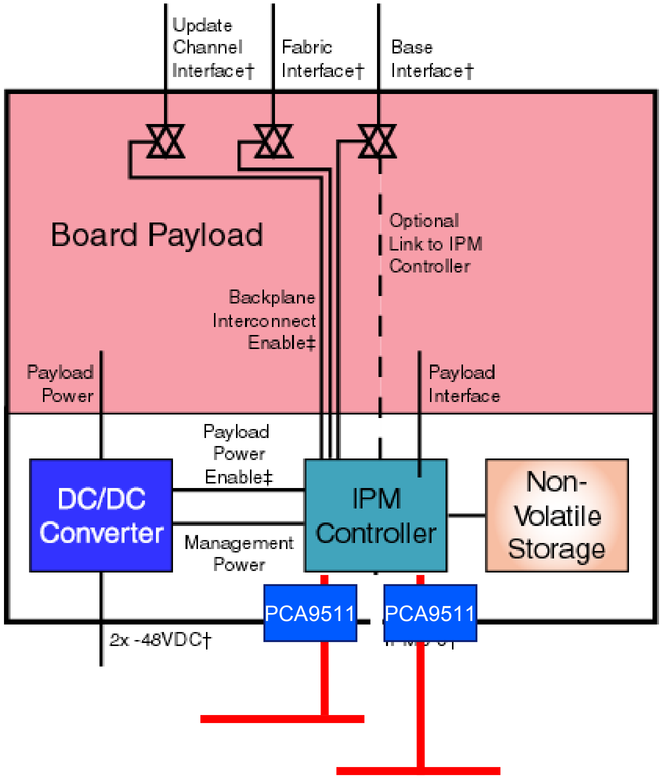
- PICMG (PCI Industrial Computer Manufacturers Group) is a consortium of over 600 companies who collaboratively develop open specifications for high performance telecommunications and industrial computing applications.
- PICMG specifications include CompactPCI® for Eurocard, rackmount applications and PCI/ISA for passive backplane, standard format cards.
- Recently, PICMG announced it was beginning development of a new series of specifications, called AdvancedTCA™, for next-generation telecommunications equipment, with a new form factor and based on switched fabric architectures
- More information - www.picmg.org

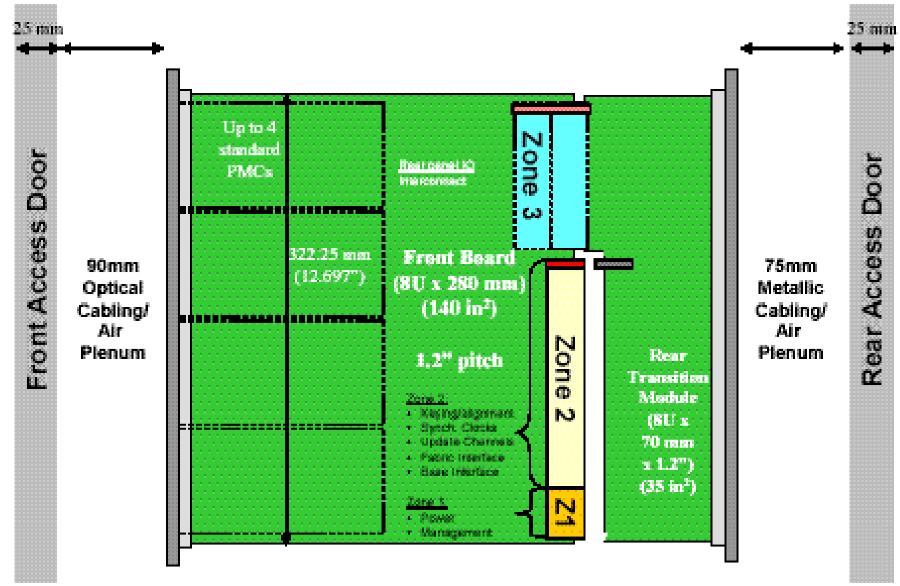
Use of IPMI within PICMG

Known as	Specification	Based on	Comments
cPCI	PICMG 2.0	NA	No IPMB
cPCI	PICMG 2.9	IPMI 1.0	Single hot swap IPMB optional
AdvancedTCA	PICMG 3.x	IPMI 1.5	Dual redundant hot swap IPMB mandatory

- PICMG 2.0: CompactPCI Core
- PICMG 2.9: System Management
- PICMG 3.0: AdvancedTCA Core
    - 3.1 Ethernet Star (1000BX and XAUI) – FC-PH links mixed with 1000BX
    - 3.2 InfiniBand® Star & Mesh
    - 3.3 StarFabric
    - 3.4 PCI Express

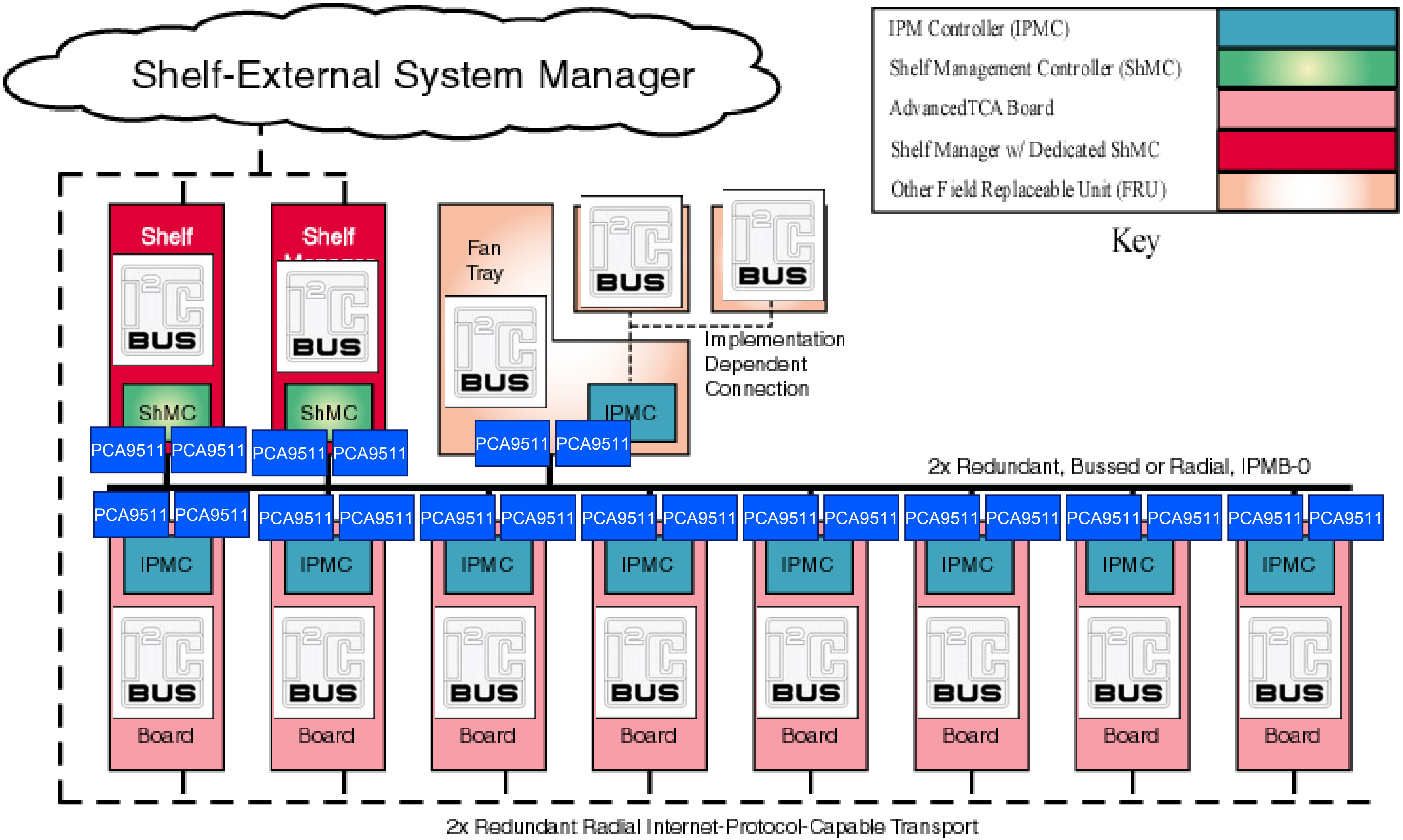
Managed ATCA Board Example





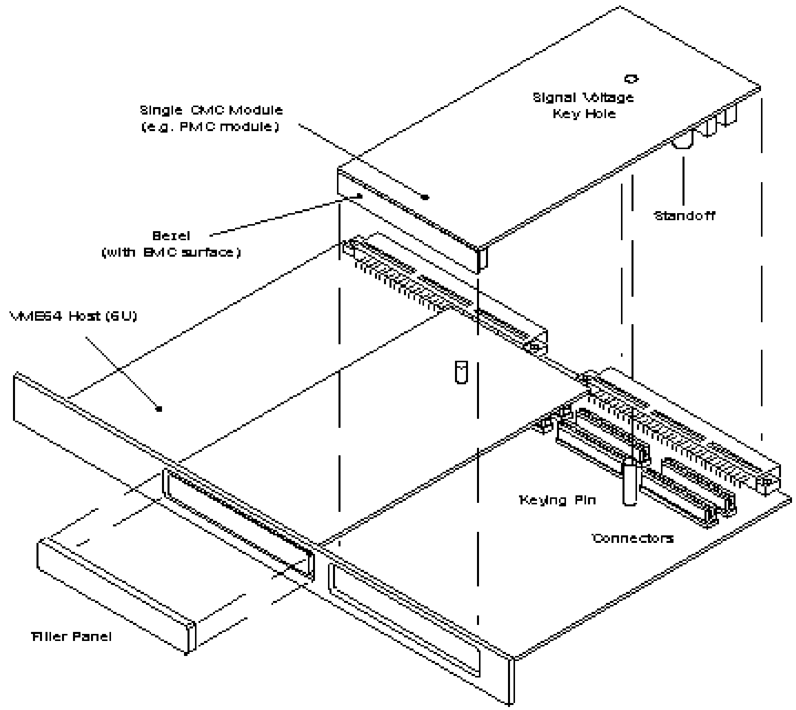
- **Dual, redundant -48VDC power distribution to each card w. high current, bladed power connector**
- **High frequency differential data connectors**
- **Robust keying block**
- **Two alignment pins**
- **Robust, redundant system management**
- **8U x 280mm card size**
- **1.2" (6HP) pitch**
- **Flexible rear I/O connector area**

Managed ATCA Shelf: Example 1



VME

- Motorola, Mostek and Signetics cooperated to define the standard
- Mechanical standard based on the Eurocard format.
    - Large body of mechanical hardware readily available
    - Pin and socket connector scheme is more resilient to mechanical wea than older printed circuit board edge connectors.
- Hundreds of component manufacturers support applications such as industrial controls, military, telecommunications, office automation and instrumentation systems.



Maximum Data Transfer Speeds		
Topology	**Bus Cycle**	**Maximum Speed**
VMEbus IEEE-1014	BLT	40 Mbyte/sec
VME64	MBLT	80 Mbyte/sec
VME64x	2eVME	160 Mbyte/sec
VME320	2eSST	320 - 500+ Mbyte/sec

www.vita.com

Use of IPMI in VME Architecture

- New VME draft standard indirectly calls for IPMI over I^{2}C for the system management protocol since there was nothing to be gained by reinventing a different form of system management for VME.
- The only change from the PICMG 2.9 system management specification is to redefine the backplane pins used for the I^{2}C bus and to redefine the capacitance that a VME board can present on the I^{2}C bus.
  - The pin change was required because the VME backplane connectors are different from cPCI.
  - The capacitance change was required because cPCI can have a maximum of 8 slots and VME can have a maximum of 21 slots.

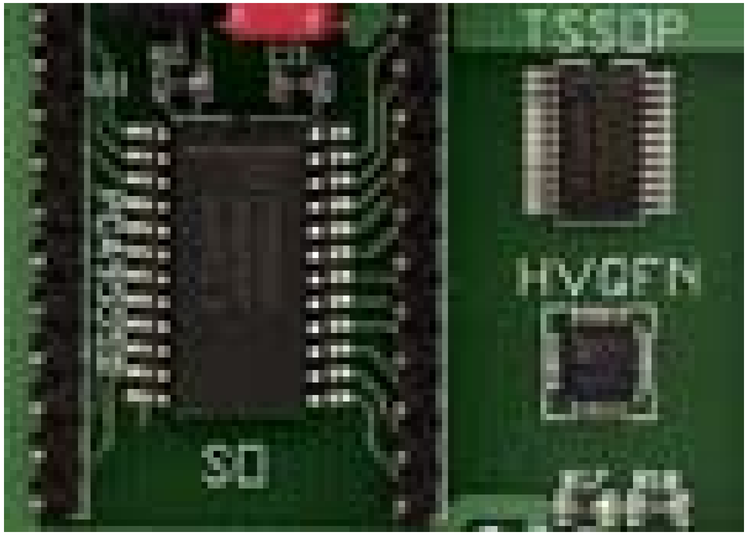
System Management for VME Draft Standard VITA 38 – 200x Draft 0.5
9 May 02 draft at www.vita.com/vso/draftstd/vita38.d0.5.pdf

I^{2}C Device Overview

I^2C Device Categories

- **TV Reception**
- **Radio Reception**
- **Audio Processing**
- **Infrared Control**
- **DTMF**
- **LCD display control**
- **Clocks/timers**
- **General Purpose I/O**
- **LED display control**
- **Bus Extension/Control**
- **A/D and D/A Converters**
- **EEPROM/RAM**
- **Hardware Monitors**
- **Microcontroller**

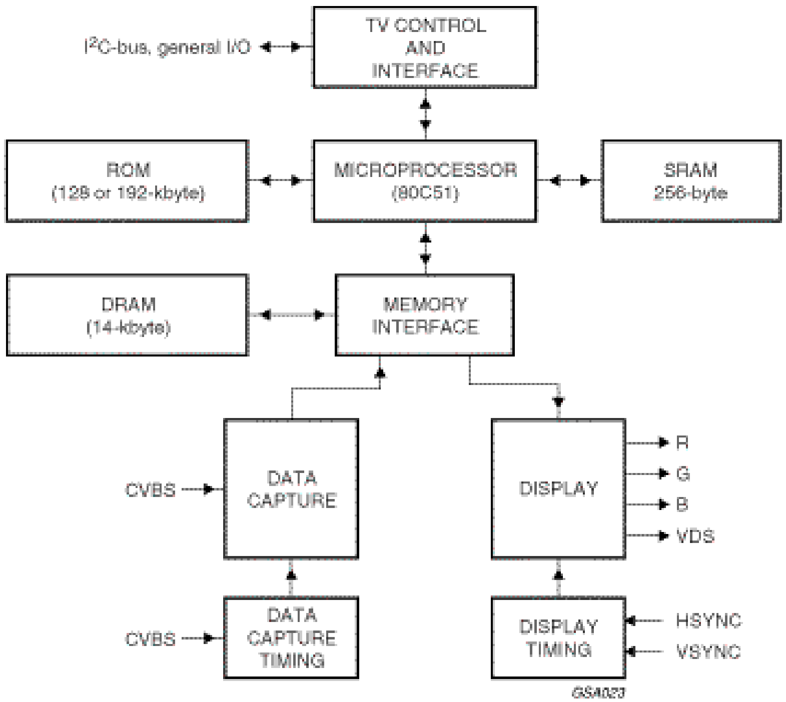
I^2C Product Characteristics



- Package Offerings
  - Typically DIP, SO, SSOP, QSOP, TSSOP or HVQFN packages
- Frequency Range
  - Typically 100 kHz operation
  - Newer devices operating up to 400 kHz
  - Graphic devices up to 3.4 MHz
- Operating Supply Voltage Range
  - 2.5 to 5.5 V or 2.8 to 5.5 V
  - Newer devices at 2.3 to 5.5 V or 3.0 to 3.6 V with 5 V tolerance
- Operating temperature range
  - Typically -40 to +85 ºC
  - Some 0 to +70 ºC
- Hardware address pins
  - Typically three (A_O, A_1, A_2) are provided to allow up to eight of the identical device on the same I^2C bus but sometimes due to pin limitations there are fewer address pins

TV Reception

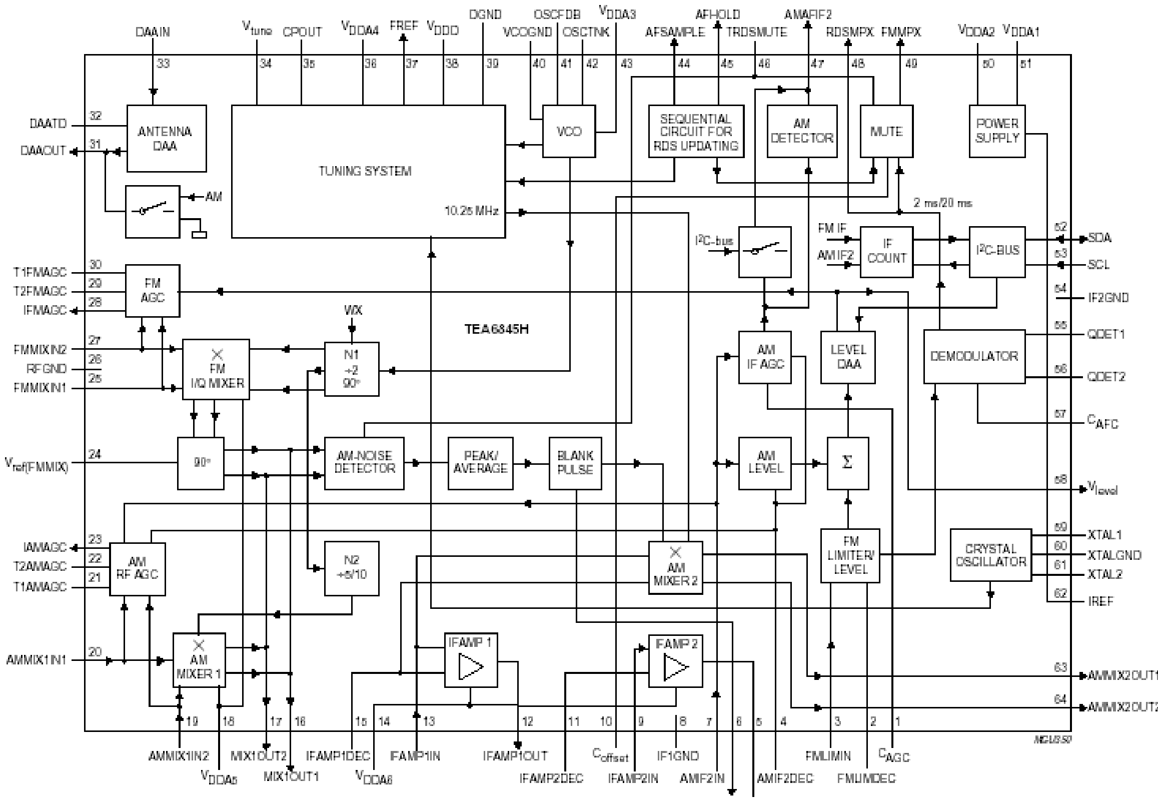
The SAA56xx family of microcontrollers are a derivative of the Philips industry-standard 80C51 microcontroller and are intended for use as the central control mechanism in a television receiver. They provide control functions for the television system, OSD and incorporate an integrated Data Capture and display function for either Teletext or Closed Caption.



Additional features over the SAA55xx family have been included, e.g. 100/120 Hz (2H/2V only) display timing modes, two page operation (50/60 Hz mode for 16:9, 4:3), higher frequency microcontroller, increased character storage, more 80C51 peripherals and a larger Display memory. For CC operation, only a 50/60 Hz display option is available.

Byte level I²C-bus up to 400 kHz dual port I/O

Radio Reception

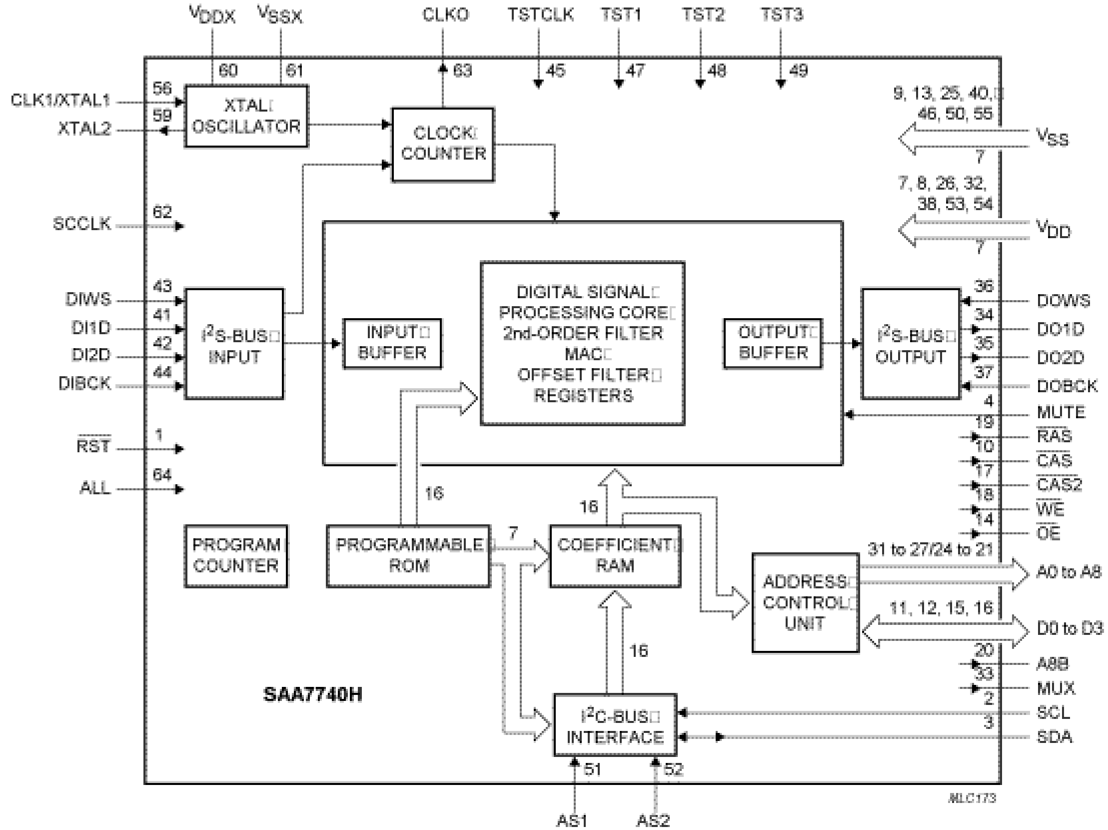


The TEA6845H is a single IC with car radio tuner for AM and FM intended for microcontroller tuning with the I²C-bus. It provides the following functions:

- AM double conversion receiver for LW, MW and SW (31 m, 41 m and 49 m bands) with IF1 = 10.7 MHz and IF2 = 450 kHz
- FM single conversion receiver with integrated image rejection for IF = 10.7 MHz capable of selecting US FM, US weather, Europe FM, East Europe FM and Japan FM bands.

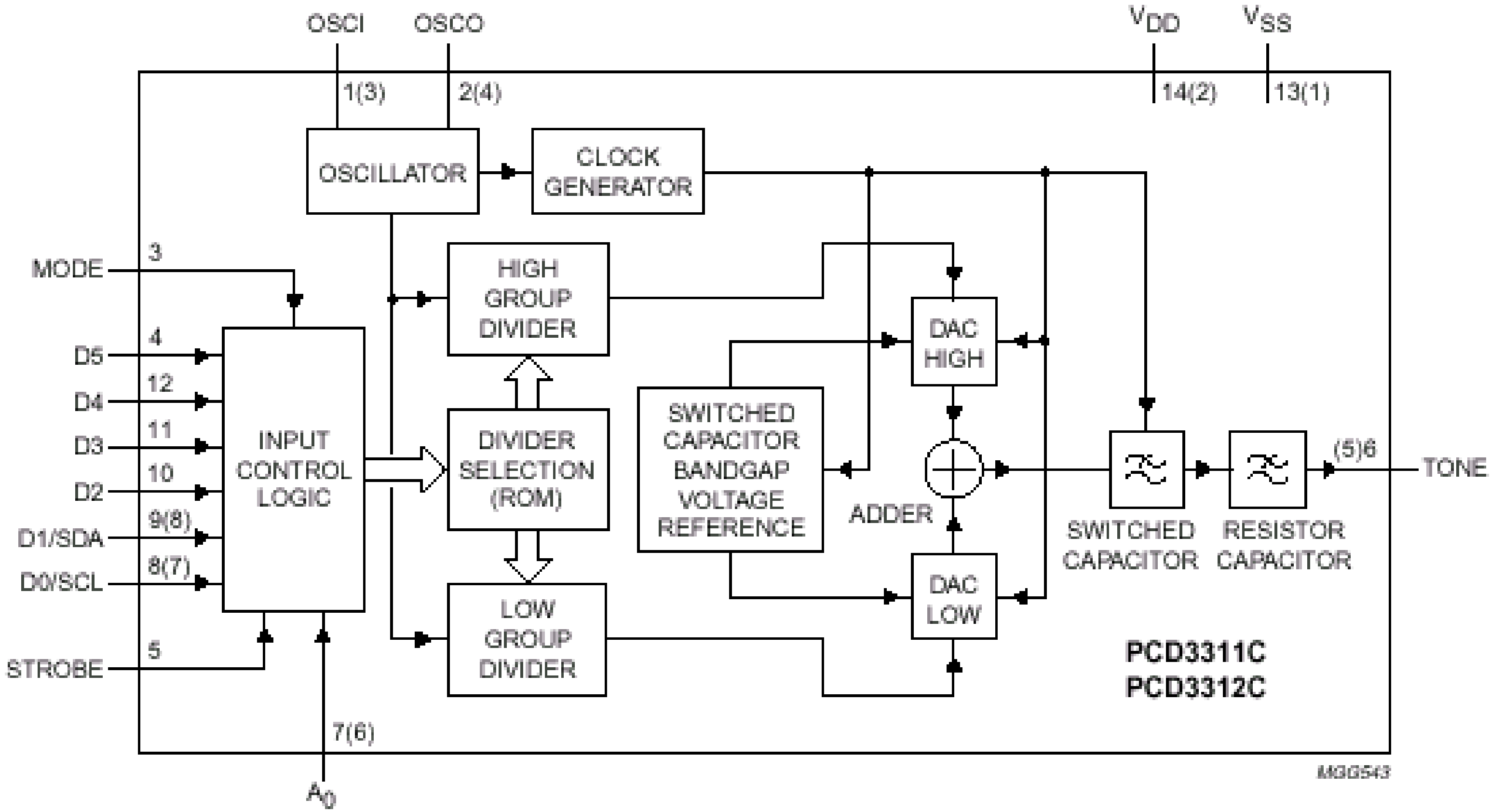
Audio Processing

The SAA7740H is a function-specific digital signal processor. The device is capable of performing processing for listening-environments such as equalization, hall-effects, reverberation, surround-sound and digital volume/balance control. The SAA7740H can also be reconfigured (in a dual and quad filter mode) so that it can be used as a digital filter with programmable characteristics.



The SAA7740H realizes most functions directly in hardware. The flexibility exists in the possibility to download function parameters, correction coefficients and various configurations from a host microcontroller. The parameters can be passed in real time and all functions can be switched on simultaneously. The SAA7740H accepts 2 digital stereo signals in the I2S-bus format at audio sampling frequency (fast) and provides 2 digital stereo outputs.

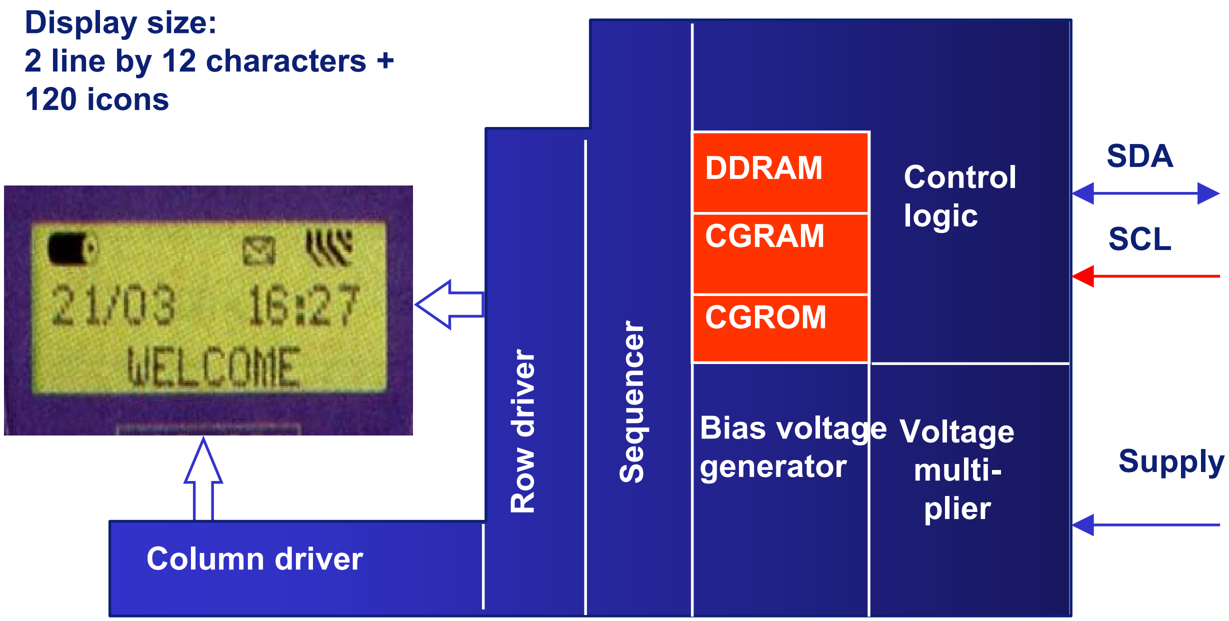
DTMF/Modem/Musical Tone Generators



- Modem and musical tone generation
- Telephone tone dialing
  - DTMF > Dual Tone Multiple Frequency
- Low baud rate modem

I^2C LCD Display Driver

LCD Display Control



The LCD Display driver is a complex device and is an example of how "complete" a system an I^2C chip can be – it generates the LCD voltages, adjusts the contrast, temperature compensates, stores the messages, has CGROM and RAM etc etc.

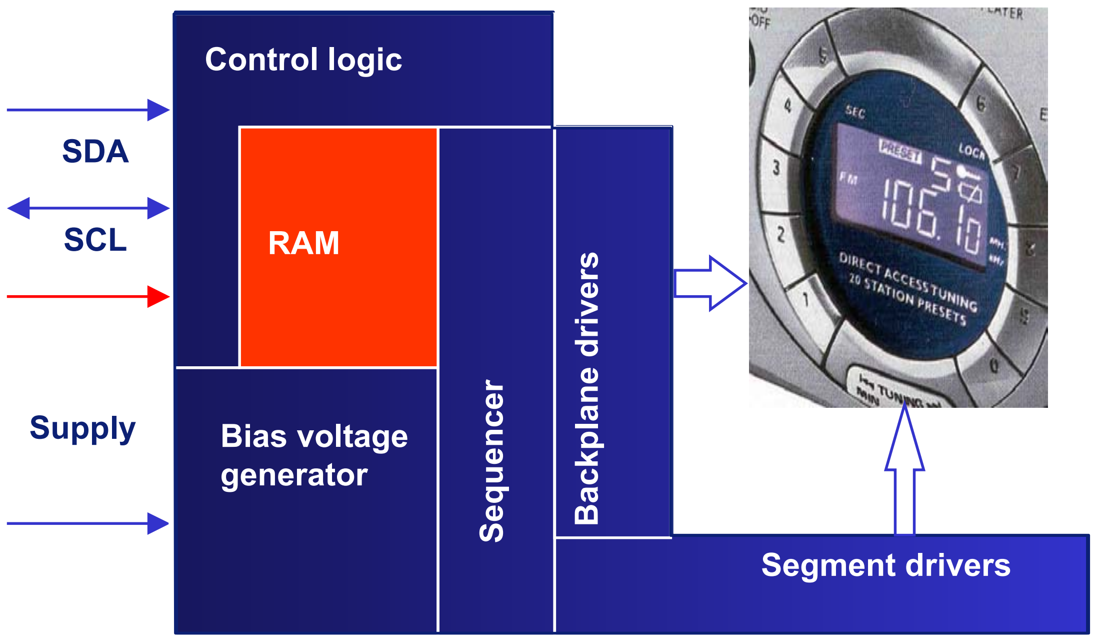
I^{2}C LCD Segment Driver

LCD Segment Control

Display sizes

1 x 24 … 2 x 40…

single chip: 4 x 40 ... 16 x 24



The LCD Segment driver is a less complex LCD driver (e.g., just a segment driver).

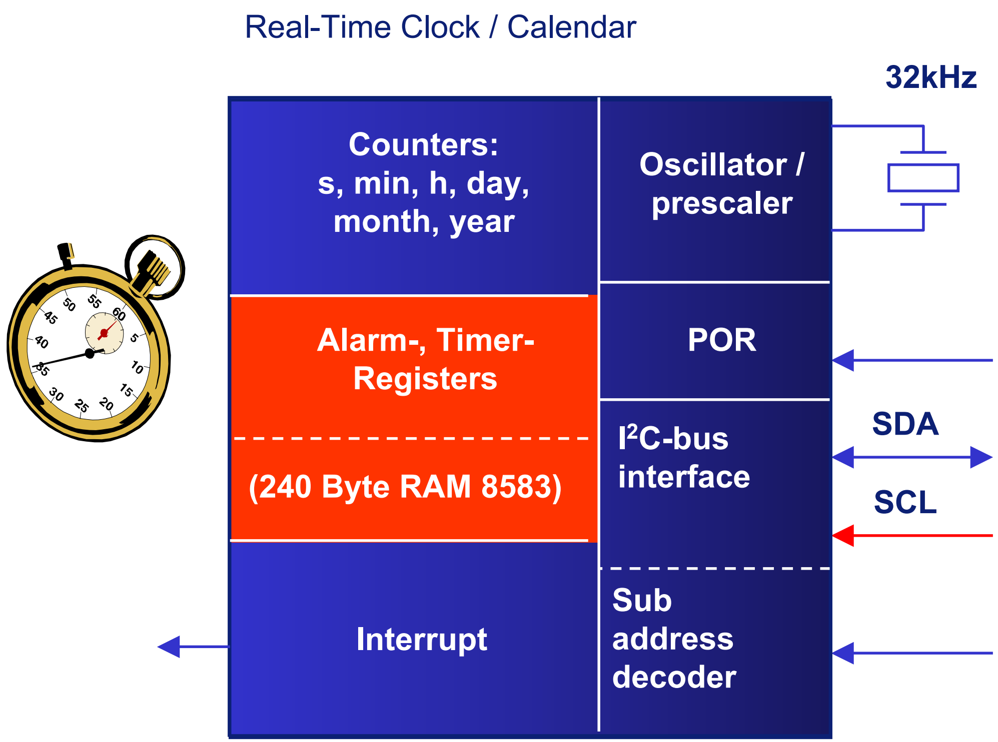
I^2C Light Sensor

The TSL2550 sensor converts the intensity of ambient light into digital signals that, in turn, can be used to control the backlighting of display screens found in portable equipment, such as laptops, cell phones, PDAs, camcorders, and GPS systems. The device can also be used to monitor and control commercial and residential lighting conditions.

By allowing display brightness to be adjusted to ambient conditions, the sensor is expected to bring about a significant reduction in the power dissipation of portables.

The TSL2550 all-silicon sensor combines two photodetectors, with one of the detectors sensitive to both visible and infrared light and the other sensitive only to IR light. The photodetectors's output is converted to a digital format, in which form the information can be used to approximate the response of the human eye to ambient light conditions sans the IR element, which the eye cannot perceive.

I^2C Real Time Clock/Calendar

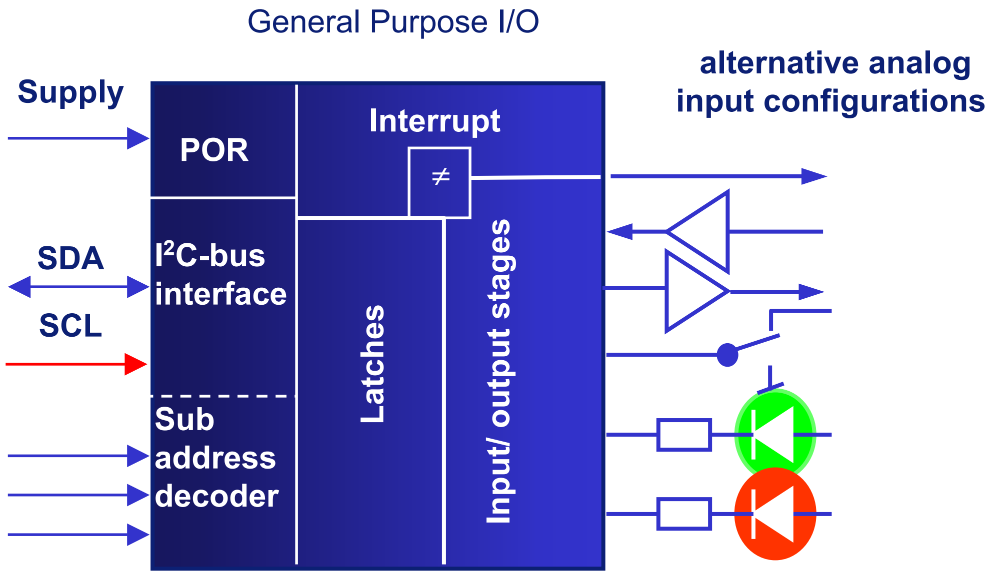


Real time clocks and event counters count the passage of time and act as a chronometer

They are used in applications such as:

- periodic alarms for safety applications
- system energy conservation
- time and date stamp for point of sales terminals or bank machines

I^{2}C General Purpose I/O Expanders



- Transfers keyboard, ACPI Power switch, keypad, switch or other inputs to microcontroller via I^{2}C bus
- Expand microcontroller via I^{2}C bus where I/O can be located near the source or on various cards
- Use outputs to drive LEDs, sensors, fans, enable and other input pins, relays and timers
- Quasi outputs can be used as Input or Output without the use of a configuration register.

Quasi Output I^{2}C I/O Expanders - Registers

- **To program the outputs**

S	Address	$\overline{W}$	A	OUTPUT DATA	A	P

Multiple writes are possible during the same communication

- **To read input values**

S	Address	$\overline{R}$	A	INPUT DATA	$\overline{A}$	P

Multiple reads are possible during the same communication

- **Important to know**
  - **At power-up, all the I/O's are HIGH; Only a current source to V_{DD} is active**
  - **An additional strong pull-up resistors allows fast rising edges**
  - **I/O's should be HIGH before using them as Inputs**

Blank

True Output I²C I/O Expanders - Registers

- **To configure the device**

S	Address	$\overline{W}$	A	03_H	A	CONFIG DATA	A	
S	Address	$\overline{W}$	A	02_H	A	POLARITY DATA	A	P

No need to access Configuration and Polarity registers once programmed

- **To program the outputs**

S	Address	$\overline{W}$	A	01_H	A	OUTPUT DATA	A	P

Multiple writes are possible during the same communication

- **To read input values**

S	Address	$\overline{W}$	A	00_H	A	S	Address	R	A	INPUT DATA	$\overline{A}$	P

Multiple reads are possible during the same communication

True Output I²C I/O Expanders - Example

Input Reg#	Polarity Reg#	Config Reg#	Output Reg#	
1	0	1	X	1
1	0	0	1	1
0	0	0	0	0
0	1	0	1	1
0	1	1	X	1
1	1	1	X	0
0	0	1	X	0
1	0	0	1	1
Read	Read/ Write	Read/ Write	Read/ Write	I/O's

Signal monitoring and/or Control

- Advantages of I^2C
  - Easy to implement (Hardware and Software)
  - Extend microcontroller: I/O’s can be located near the source or on various cards
  - Save GPIO’s in the microcontroller
  - Only 2 wires needed, independently of the numbers of signals
  - Signal(s) can be far from the masters
  - Fast enough to control keyboards
  - Simplify the PCB layout
  - Devices exist in the market and are massively used

Signal monitoring and/or Control

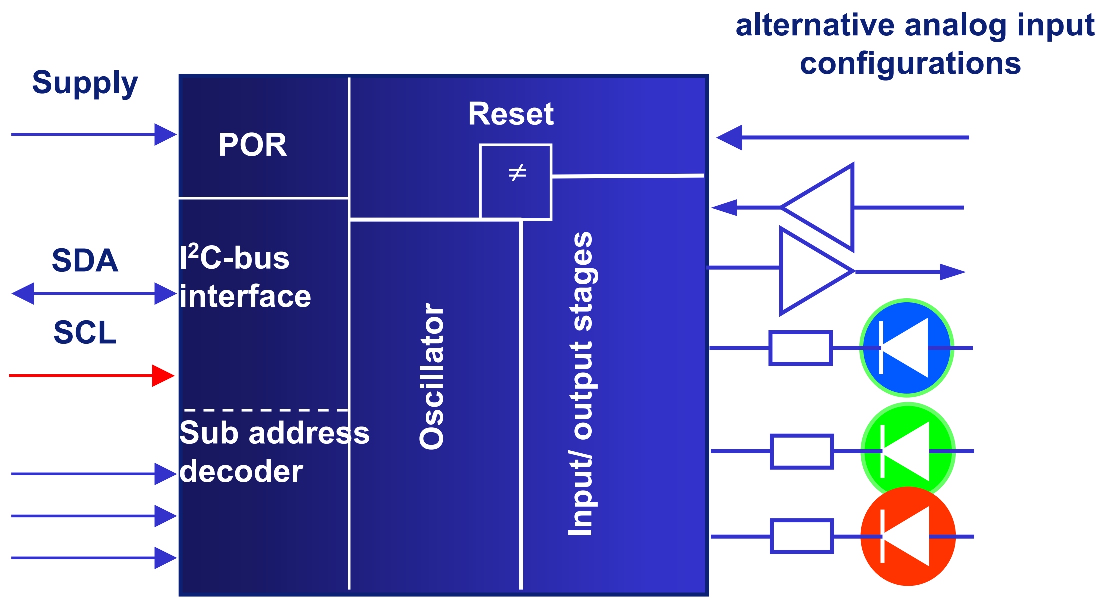
- Proposed devices

# of Outputs	Interrupt and POR	POR and 2K EEPROM	Interrupt, POR and 2K EEPROM
Quasi Output (20-25 ma sink and 100 uA source)			
8	PCF8574/74A	PCA9500/58	PCA9501
16	PCF8575/75C	-	-

# of Outputs	Reset and POR	Interrupt and POR
True Output (20-25 ma sink and 10 mA source)		
8	PCA9556/57	PCA9534/54/54A
16	-	PCA9535/55

- Advantages
  - Number of I/O scalable
  - Programmable I^2C address allowing more than one device in the bus
  - Interrupt output to monitor changes in the inputs
  - Software controlling the device(s) easy to implement

I²C LED Dimmers and Blinkers

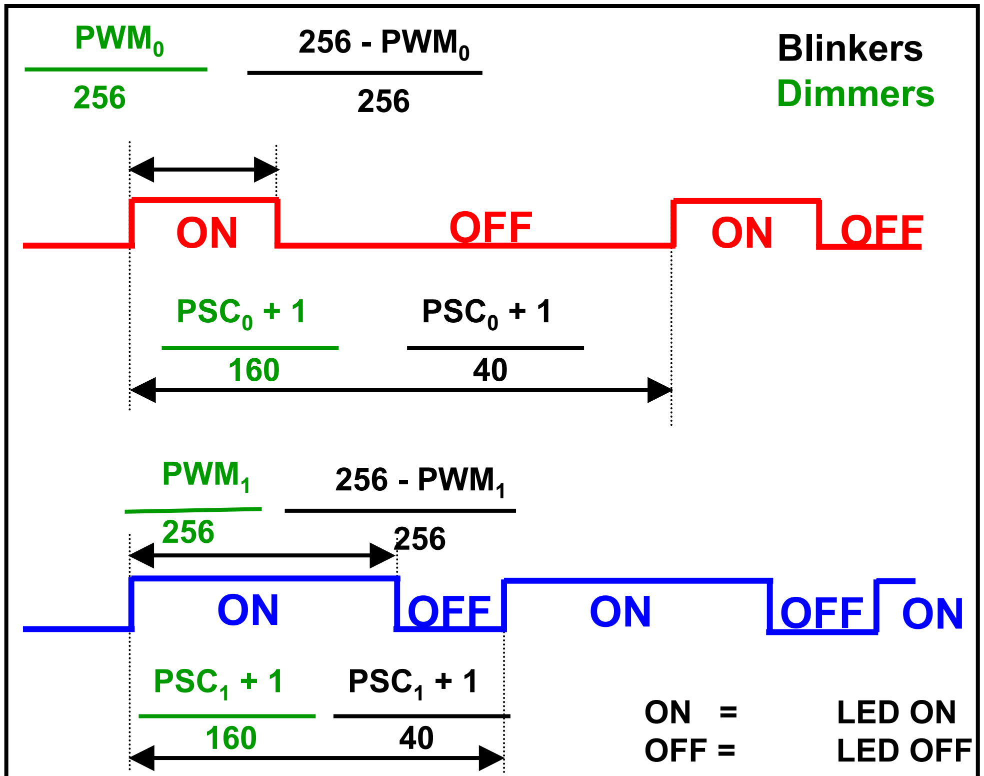


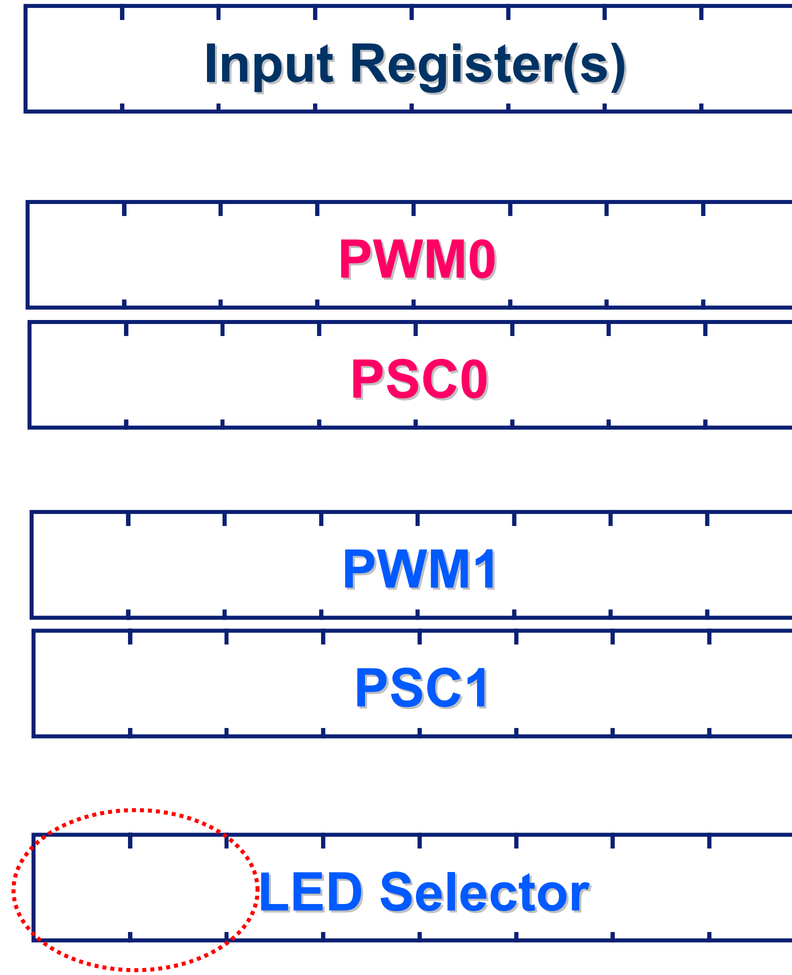
- I²C/SMBus is not tied up by sending repeated transmissions to turn LEDs on and then off to “blink” LEDs.
- Frees up the micro’s timer
- Continues to blink LEDs even when no longer connected to bus master
- Can be used to cycle relays and timers
- Higher frequency rate allows LEDs to be dimmed by varying the duty cycle for Red/Green/Blue color mixing applications.

I^{2}C LED Blinkers and Dimmers

	0 (00$_H$)	255 (FF$_H$)
Frequency	40 Hz	6.4 s
Duty Cycle	100 %	0.4 %

	0 (00$_H$)	255 (FF$_H$)
Frequency	160 Hz	1.6 s
Duty Cycle	0 %	99.6 %





ON, OFF, BR1, BR2

I^2C Blinkers and Dimmers - Programming

- **To program the 2 blinking rates**

PSC0 pointer = 01_H for 2, 4 and 8-bit devices
PSC0 pointer = 02_H for the 16-bit devices

- **To program the drivers**

LEDSEL0 pointer = 05_H for 2, 4 and 8-bit devices

LEDSEL0 pointer = 06_H for the 16-bit devices

Only the 16-bit devices have 4 LED selector registers (8-bit devices have 2 registers, 2 and 4-bit devices have only one)

Using I²C for visual status

- Use LEDs to give visual interpretation of a specific action:
  - alarm status (using different blinking rates)
  - battery charging status
- 1st approach: I²C GPIO's
  - Advantage:
    - Simple programming
    - Easy to implement
  - Inconvenient:
    - Need to continually send ON/OFF commands through I²C
    - 1 microcontroller's timer required to perform the task
    - I²C bus can be tied up by commands if many LEDs to be controlled
    - Blinking is lost if the I²C bus hangs
- 2nd approach: I²C LED Blinkers
  - Advantage:
    - One time programmable (frequency, duty cycle)
    - Internal oscillator
    - Easy to implement
    - Device does not need I²C bus once programmed and turned on

Using I²C for visual status

- Products:

# of Outputs	Reset and POR
2	PCA9550
4	PCA9553
8	PCA9551
16	PCA9552

LED Blinkers

Blinking between 40 times a second to once every 6.4 seconds

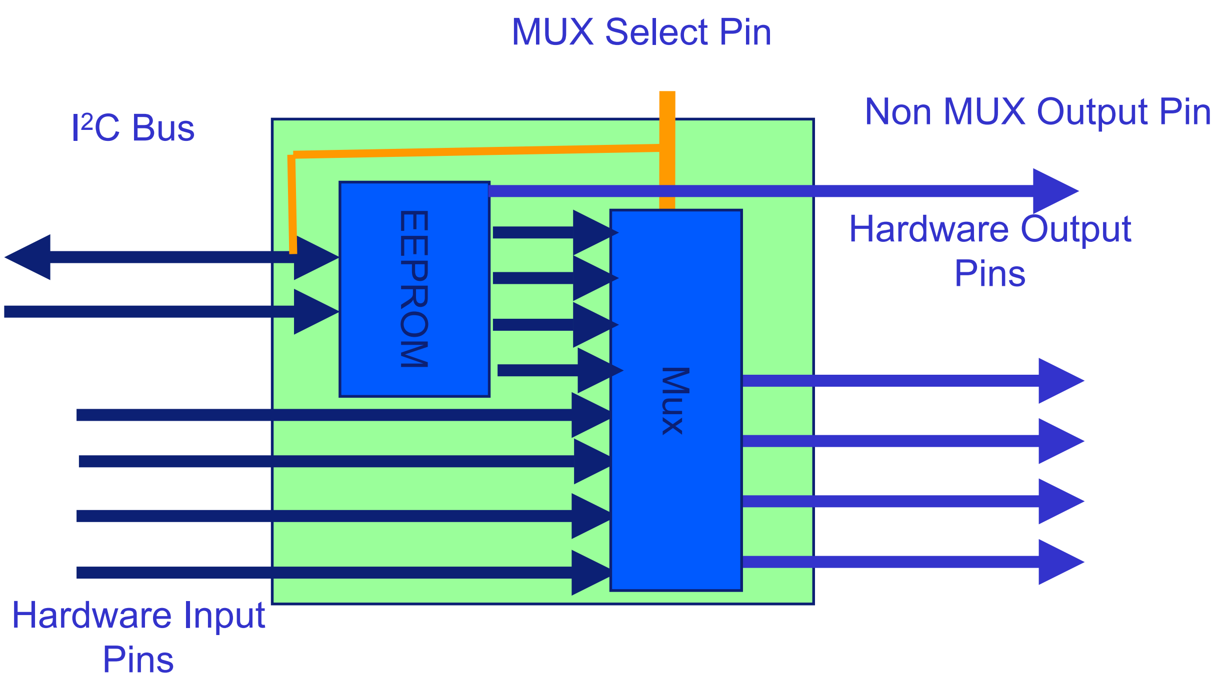
# of Outputs	Reset and POR
2	PCA9530
4	PCA9533
8	PCA9531
16	PCA9532

LED Dimmers

Blinking between 160 times a second to once every 1.6 seconds.

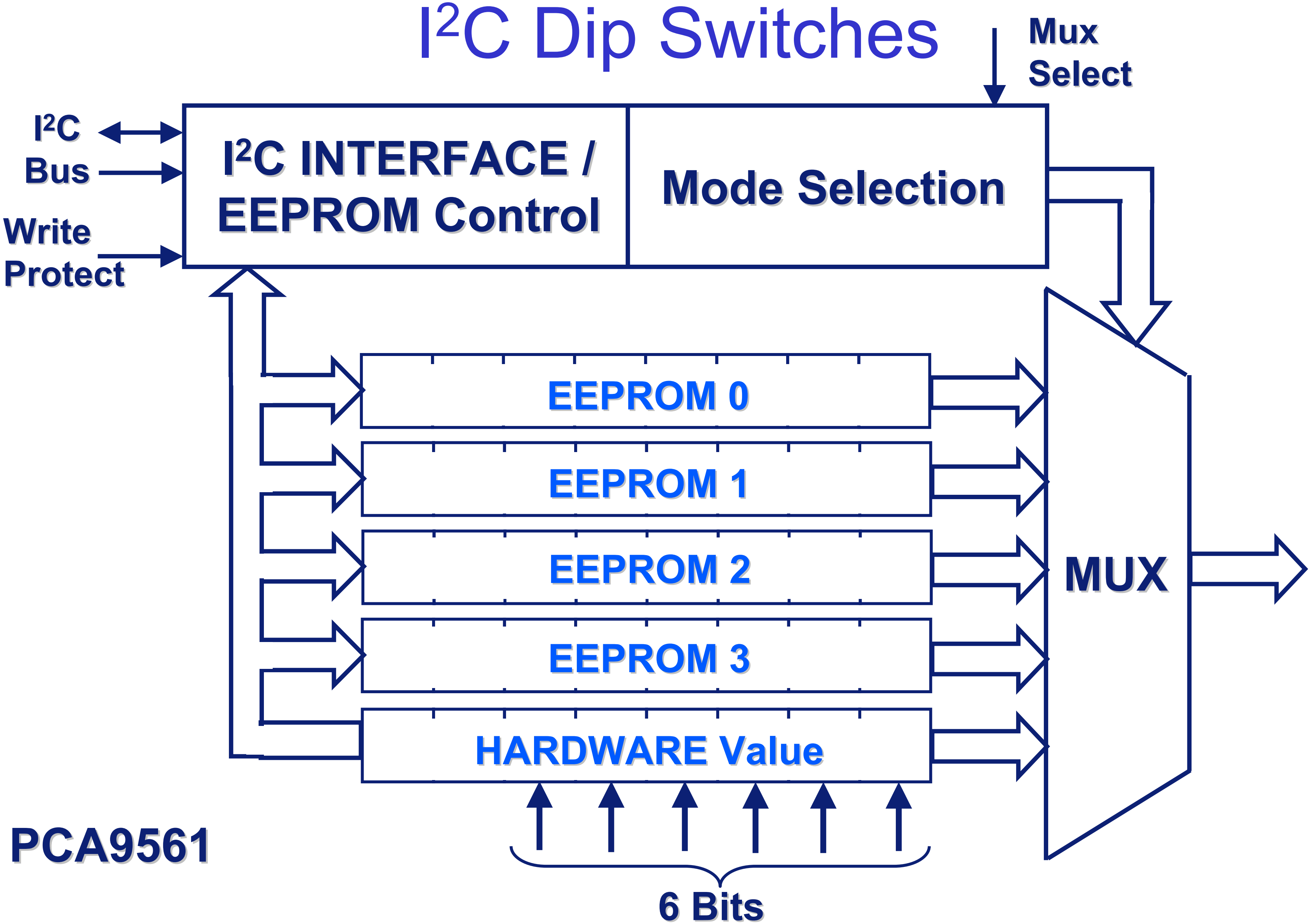
Can be used for dimming/brightness or PWM for stepper motor control

I^2C DIP Switches



- Non-volatile EEPROM retains values when the device is powered down
- Used for Speed Step™ notebook processor voltage changes when on AC/battery power or when in deep sleep mode
- Also used as replacement for jumpers or DIP switches since there is no requirement to open the equipment cabinet to modify the jumpers/DIP switch settings

I²C Dip Switches



PCA9561

I^{2}C DIP Switches - PCA9561

- **To program the 4 EEPROMS**

S	Address	$\overline{W}$	A	00_H	A	EEPROM 0	A	EEPROM 1	A

A	EEPROM 2	A	EEPROM 3	A	P

- **To read the 4 EEPROMS**

S	Address	$\overline{W}$	A	00_H	A	S	Address	R	A	EEPROM 0	A

EEPROM 1	A	EEPROM 2	A	EEPROM 3	$\overline{A}$	P

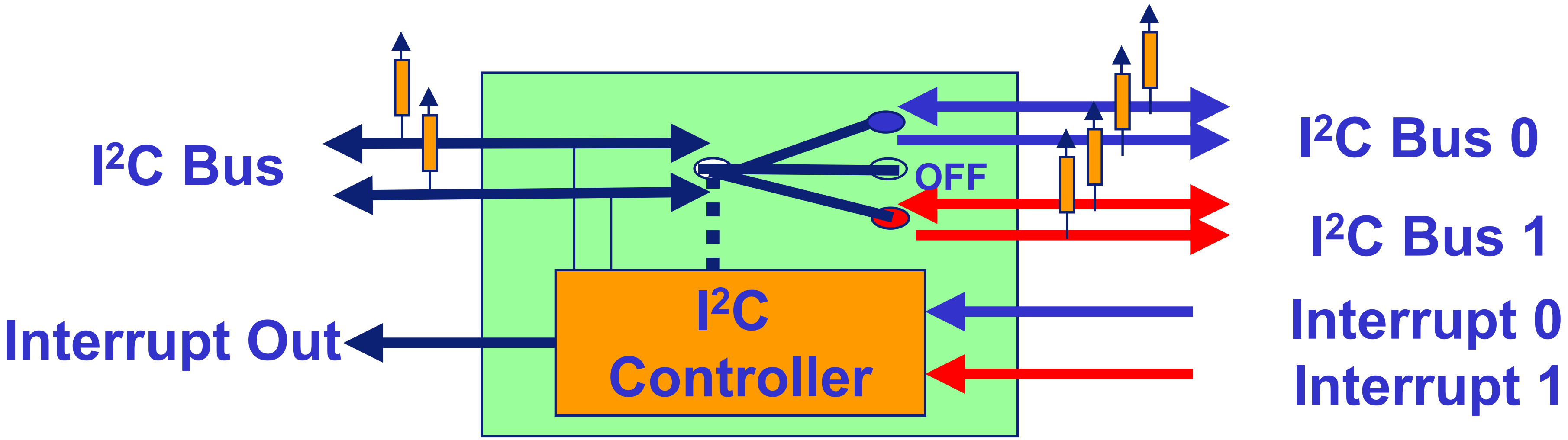
- **To read the Hardware value**

S	Address	$\overline{W}$	A	FF_H	A	S	Address	R	A	HW VALUE	$\overline{A}$	P

- **To select the mode**

S	Address	$\overline{W}$	A	FX_H	A	P

I^{2}C Multiplexers



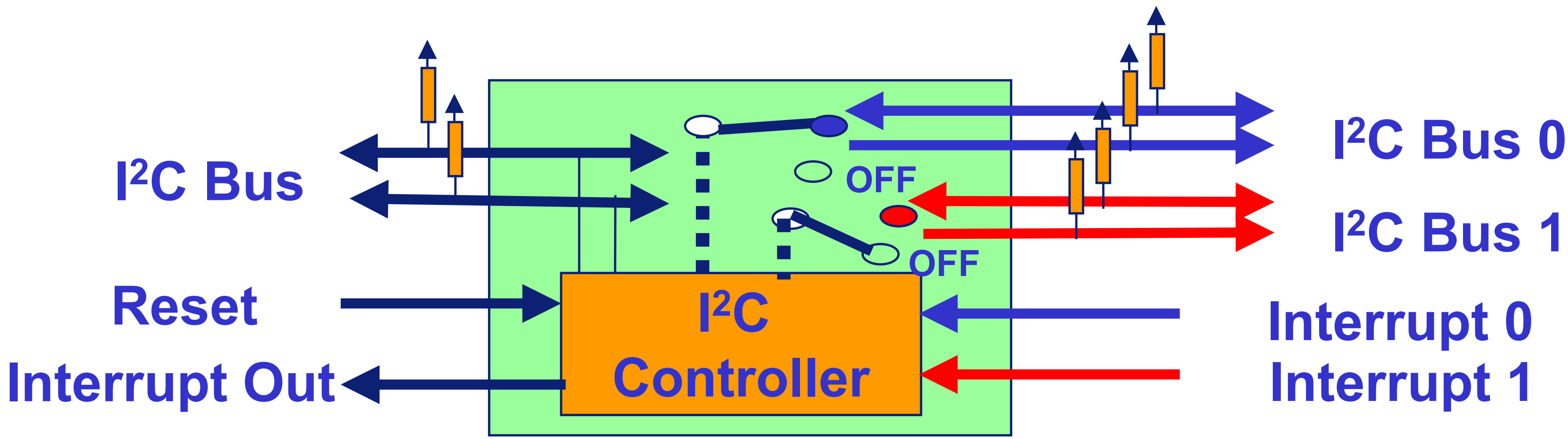
FEATURES

-Fan out main I^{2}C/SMBus to multiple channels
-Select off or individual downstream channel
-I^{2}C/SMBus commands used to select channel
-Power On Reset (POR) opens all channels
-Interrupt logic provides flag to master for system monitoring.

KEY POINTS

-Many specialized devices have only one I^{2}C address and sometimes many are needed in the same system.
-Multiplexers allow the master to communicate to one downstream channel at a time but don't isolate the bus capacitance
-Other Applications include sub-branch isolation.

I^{2}C Switches



- Switches allow the master to communicate to one channel or multiple downstream channels at a time
- Switches don't isolate the bus capacitance
- Other Applications include: sub-branch isolation and I^{2}C/SMBus level shifting (1.8, 2.5, 3.3 or 5.0 V)

I²C Multiplexers & Switches - Programming

- **To connect the upstream channel to the selected downstream channel(s)**

S	PCA954x Address	$\overline{W}$	A	CHANNEL SELECTION	A	P

Selection is done at the STOP command

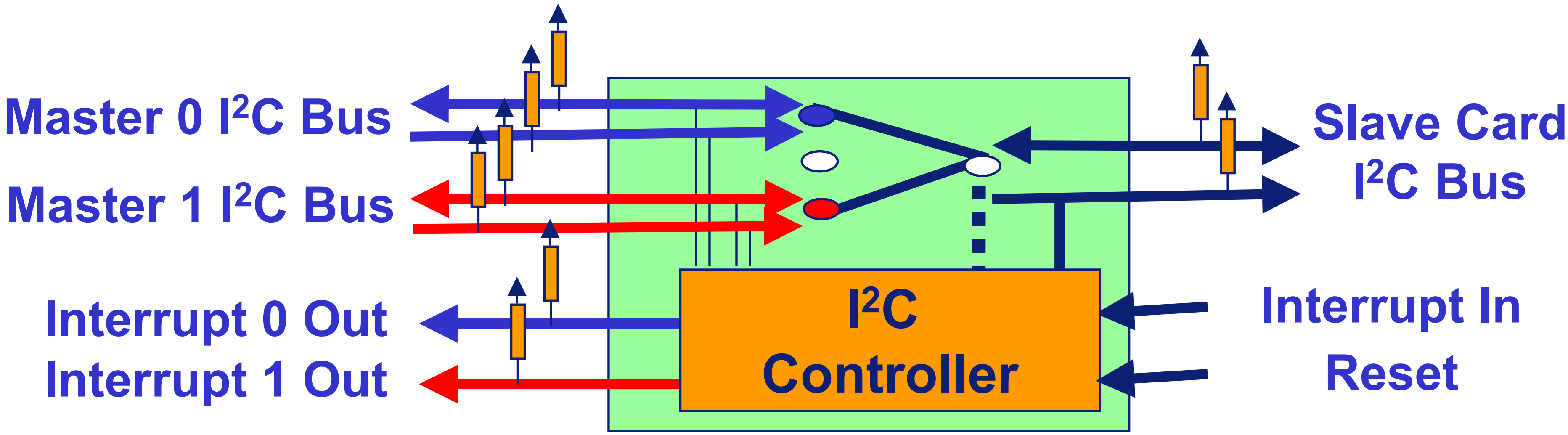
- **To access the downstream devices on the selected channel**

S	Device Address	$\overline{W}$	A	Command	A	●●●●	P

Once the downstream channel selection is done, there is no need to access (Write) the PCA954x Multiplexer or Switch

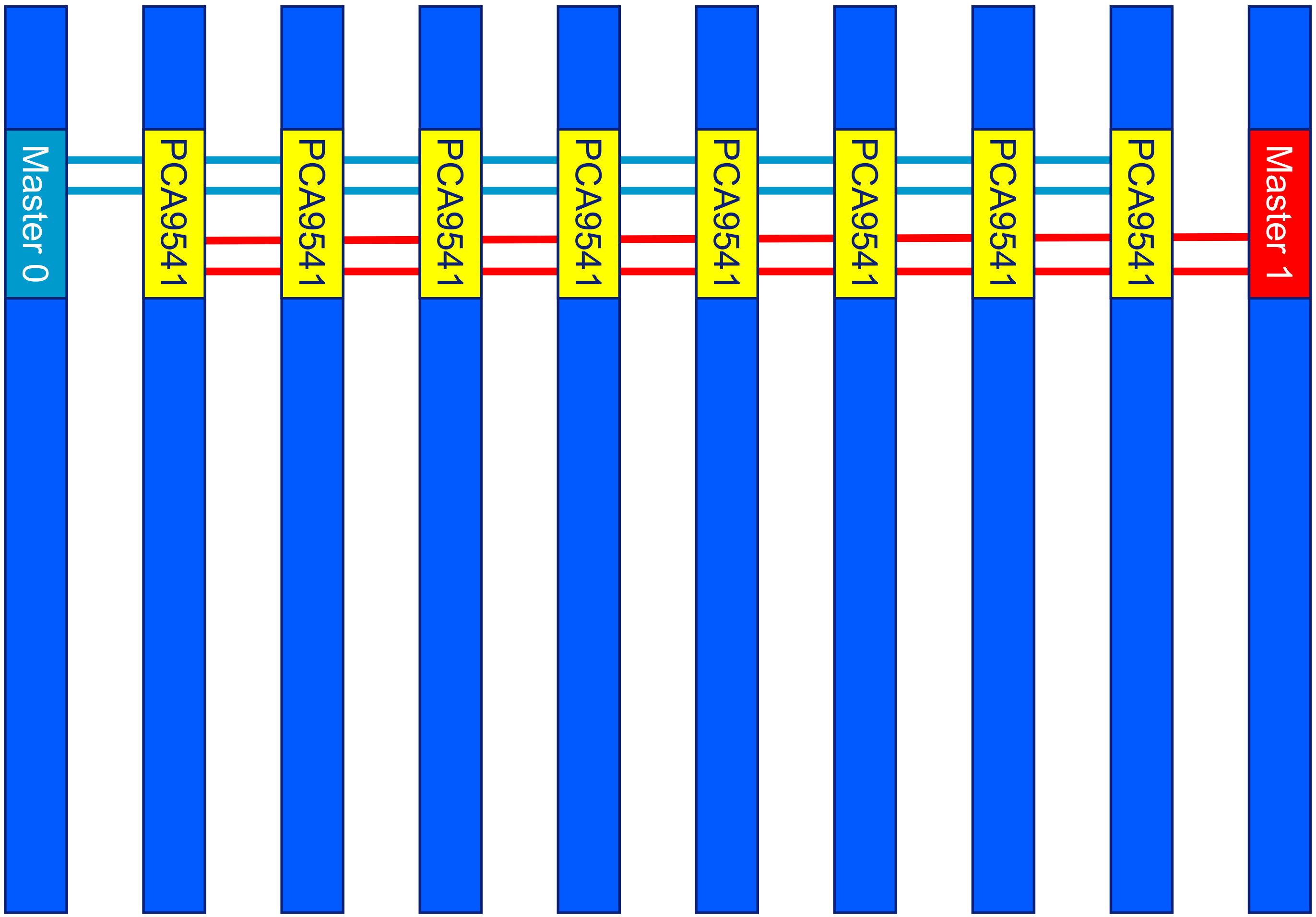
The device will keep the configuration until a new configuration is required (New Write operation on the PCA954x)

I^2C 2 to 1 Master Selector

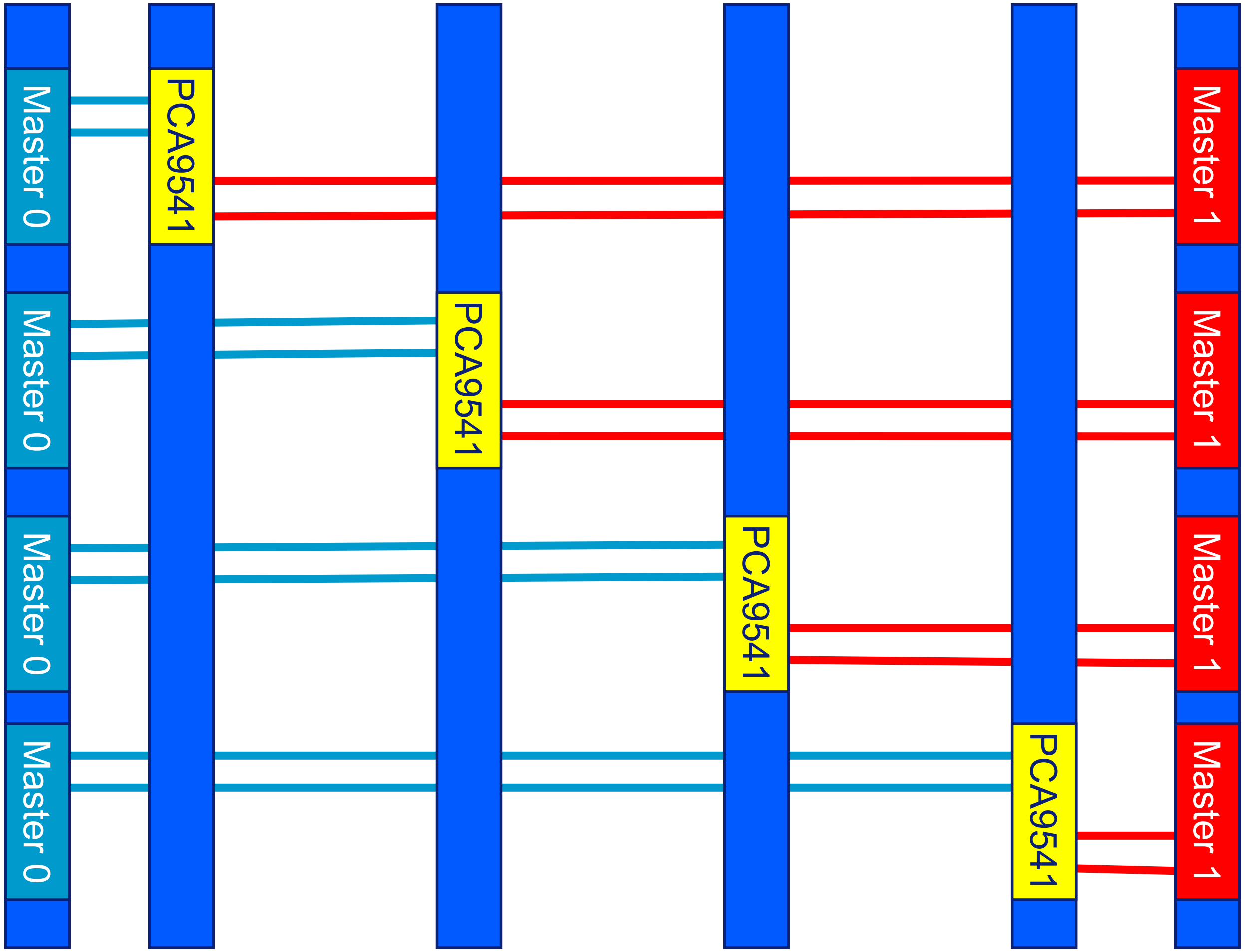


- Master Selector selects from two I^2C/SMBus masters to a single channel
- I^2C/SMBus commands used to select master
- Interrupt outputs report demultiplexer status
- Sends 9 clock pulses/stop to clear slaves prior to transferring master

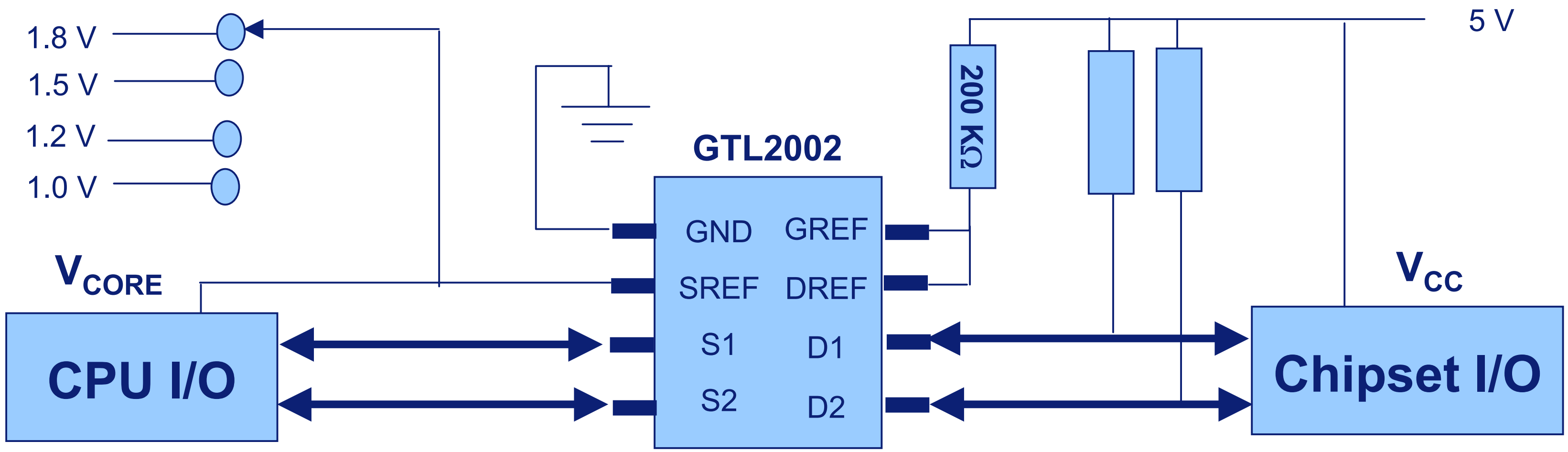
Master Selector in Multi-Point Application



Master Selector in Point-Point Application

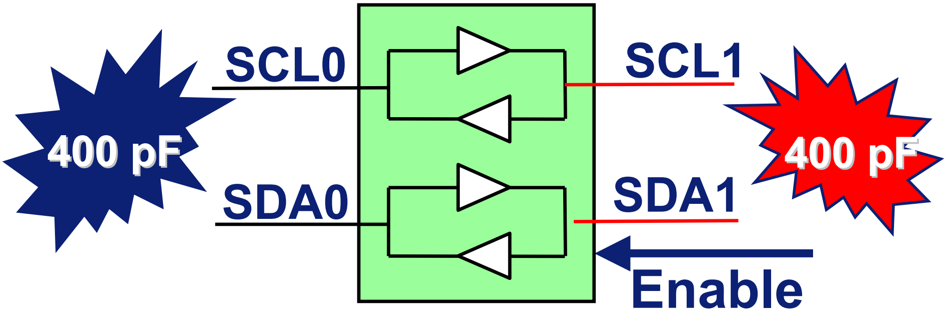


I^2C Bus Bi-Directional Voltage Level Translation



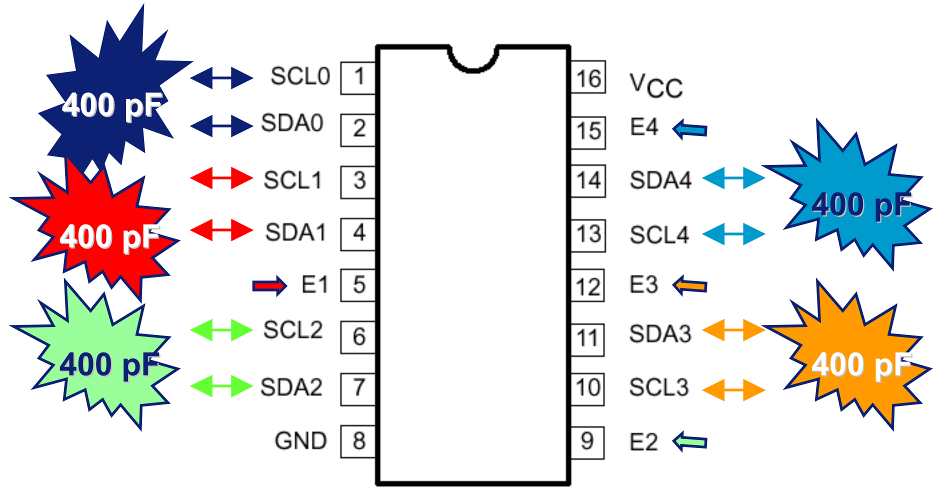
- Voltage translation between any voltage from 1.0 V to 5.0 V
- Bi-directional with no direction pin
- Reference voltage clamps the input voltage with low propagation delay
- Used for bi-directional translation of I^2C buses at 3.3 V and/or 5 V to the processor I^2C port at 1.2 V or 1.5 V or any voltage in-between
- BiCMOS process provides excellent ESD performance

I^{2}C Bus Repeater and Hub



I^{2}C Bus Repeater

PCA9515

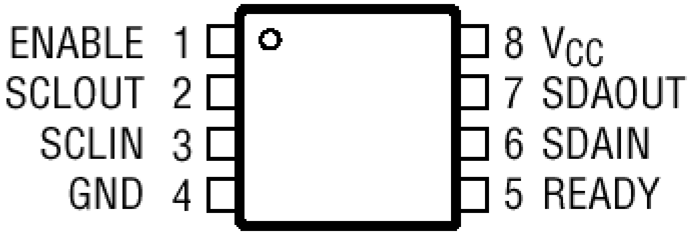


5-Channel I^{2}C Hub

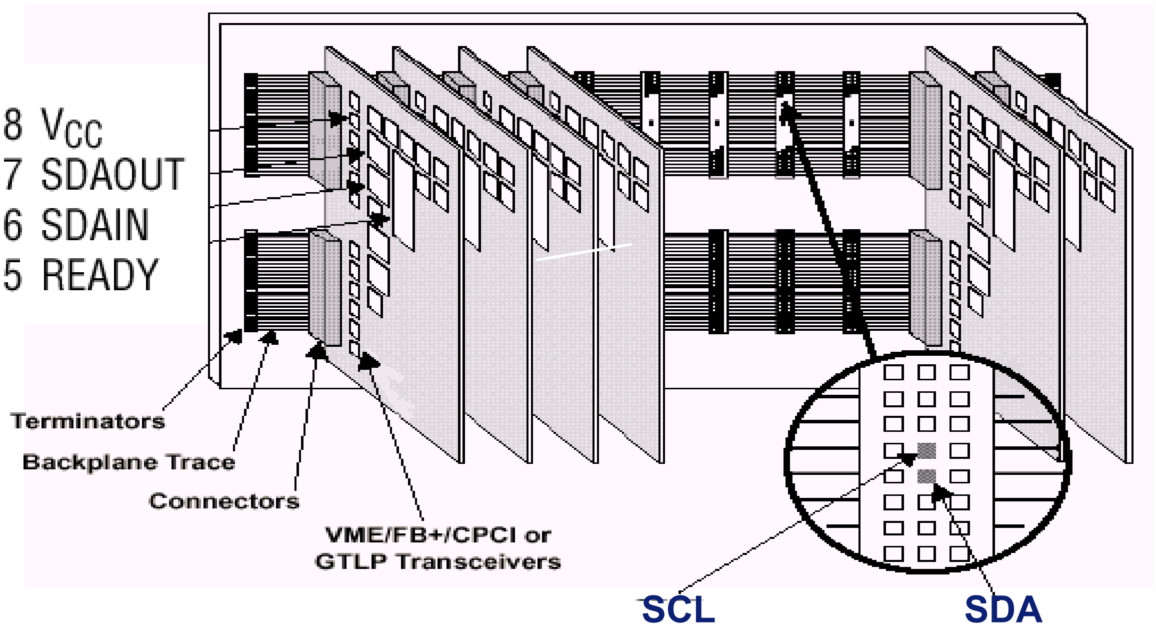
PCA9516

- Bi-directional I^{2}C drivers isolate the I^{2}C bus capacitance to each segment.
- Multi-master capable (e.g., repeater transparent to bus arbitration and contention protocols) with only one repeater delay between segments.
- Segments can be individually isolated
- Voltage Level Translation
    - 3.3 V or 5 V voltage levels allowed on the segment

I^{2}C Hot Swap Bus Buffer

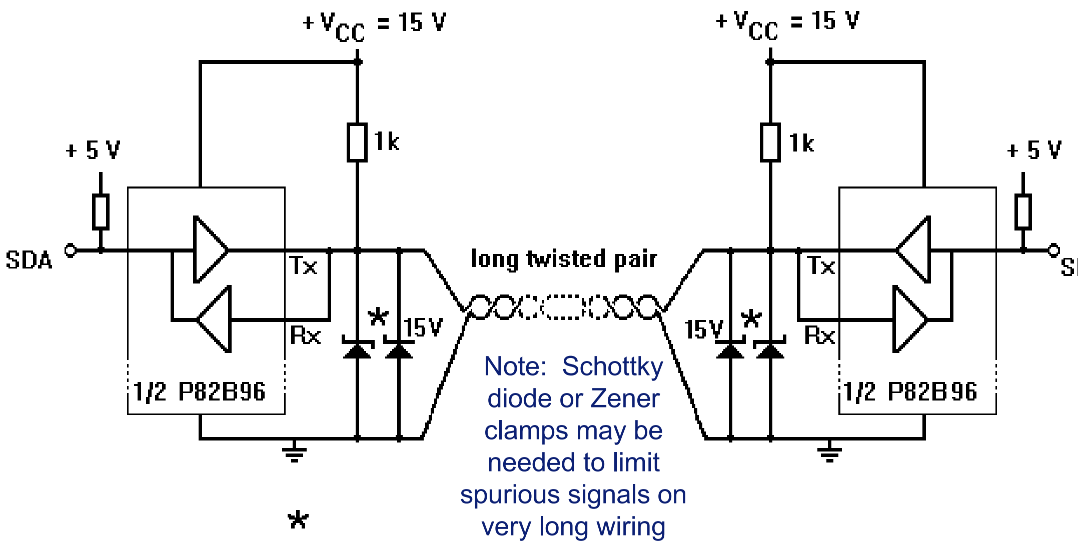


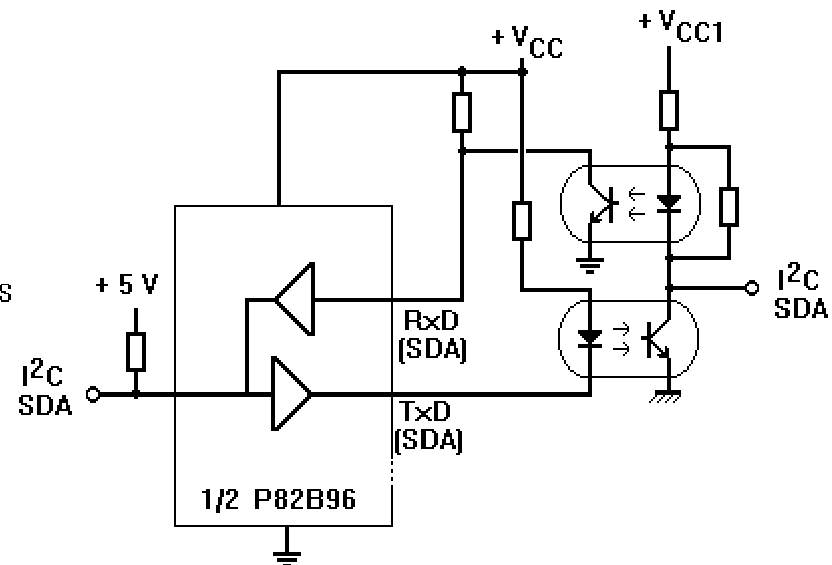
PCA9511
PCA9512
PCA9513
PCA9514

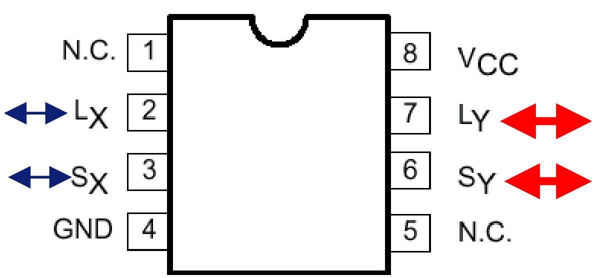


- Allows I/O card insertion into a live backplane without corruption of busses
- Control circuitry connects card after stop bit or idle occurs on the backplane
- Bi-directional buffering isolates capacitance, allows 400 pF on either side
- Rise time accelerator allows use of weaker DC pull-up currents while still meeting rise time requirements
- SDA and SCL lines are precharged to 1V, minimizing current required to charge chip parasitic capacitance

I^{2}C Bus Extenders







I^{2}C Bus Extender

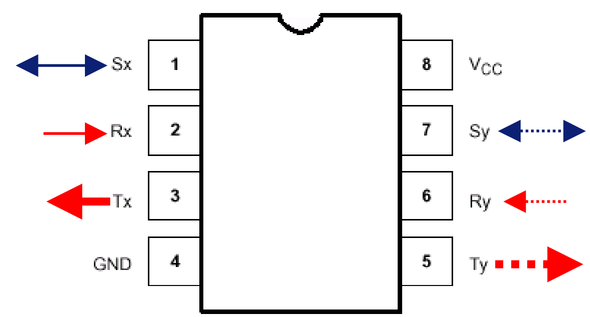
P82B715

KEY POINTS

High drive outputs are used to extend the reach of the I2C bus and exceed the 400 pF/system limit.

Possible distances range from 50 meters at 85kHz to 1km at 31kHz over twisted-pair phone cable.

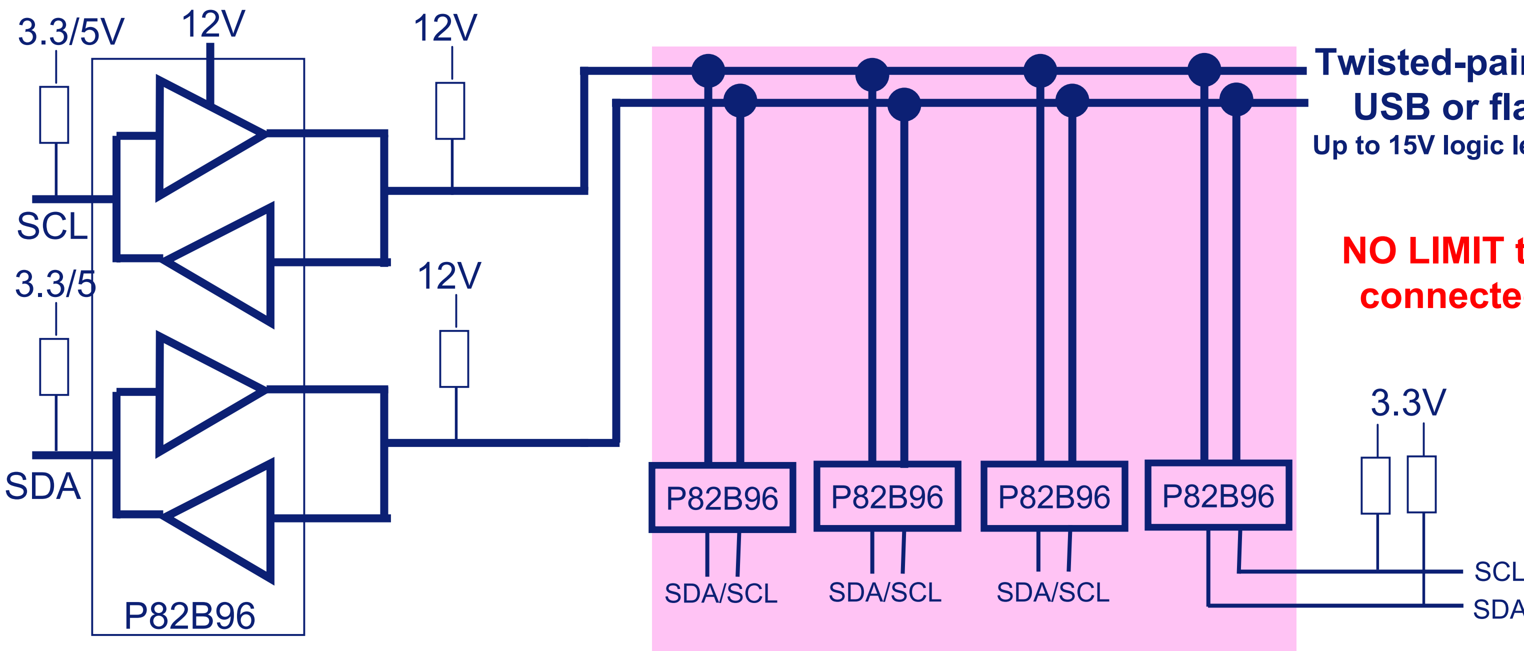
Bus Buffer has split high drive outputs allowing differential transmission or Opto-isolation of the I2C Bus.



Dual Bi-Directional Bus Buffer

P82B96

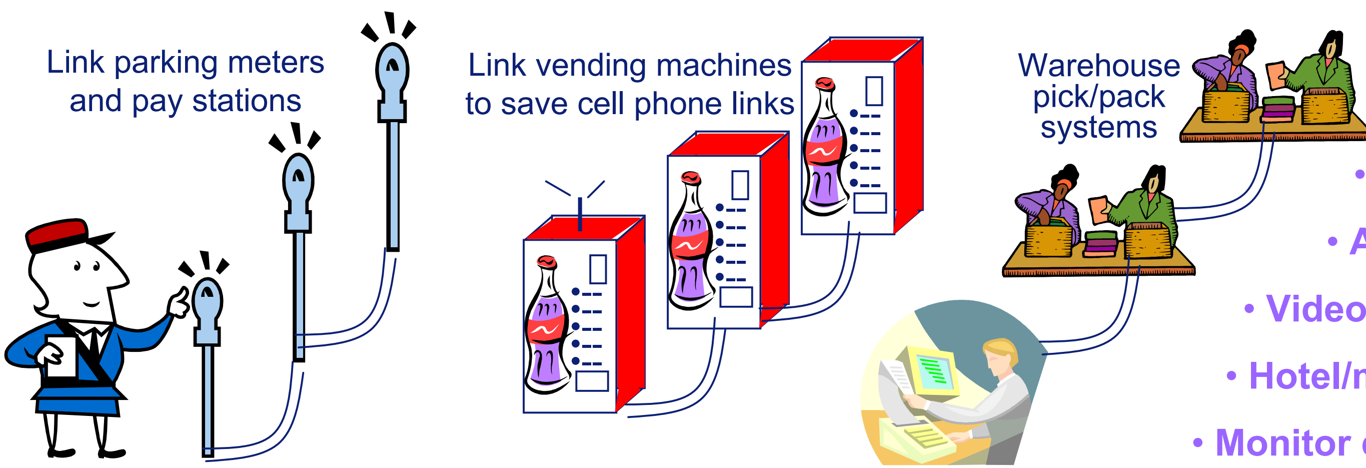
Changing I²C bus signals for multi-point applications



Twisted-pair telephone wires, USB or flat ribbon cables

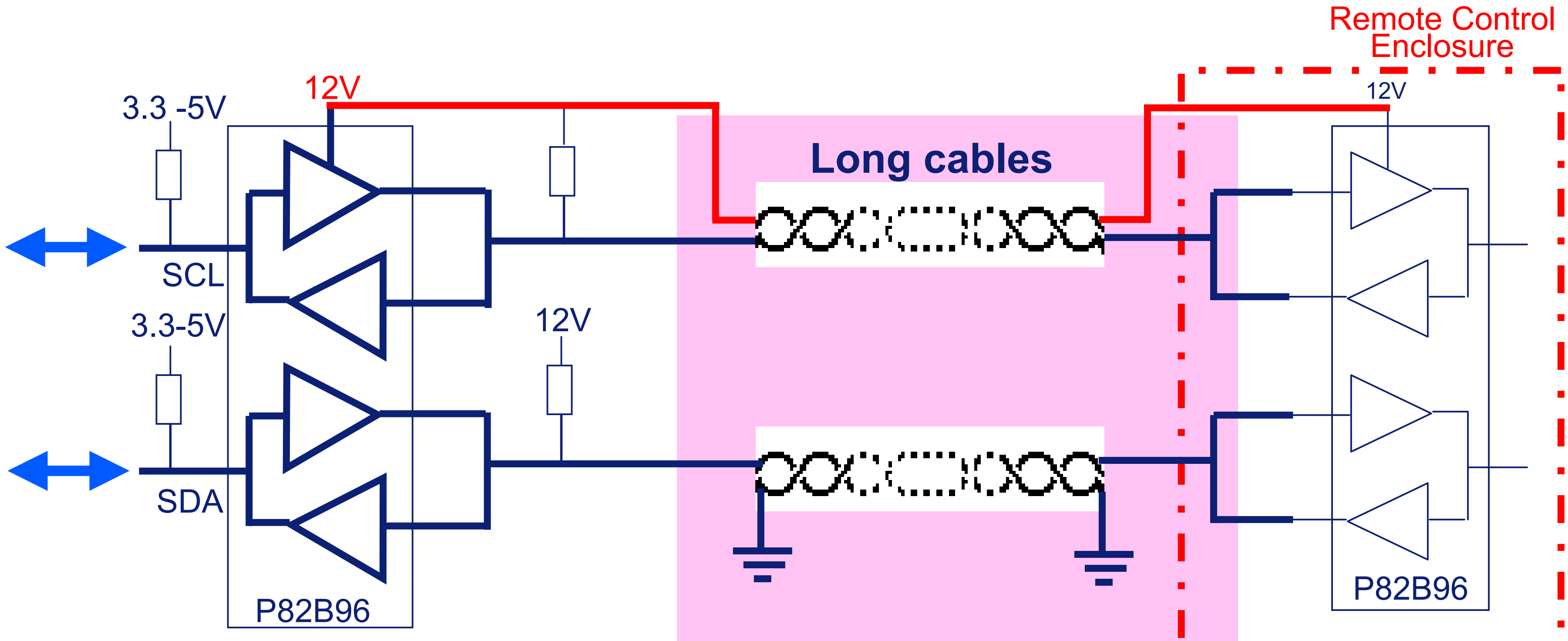
Up to 15V logic levels, Include V_{CC} & GND

NO LIMIT to the number of connected bus devices !



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

Changing I^2C bus signals for driving long distances



Bi-directional data streams

Special logic levels (I^2C compatible 5V)

I^2C currents (3mA)

Simply link the pins for Bi-directional data streams

Conventional CMOS logic levels (2-15V)

Higher current option, up to 30mA static sink

Twisted-pair telephone wires, USB or flat ribbon cables

2V through 12V logic levels

Able to send V_{CC} and GND

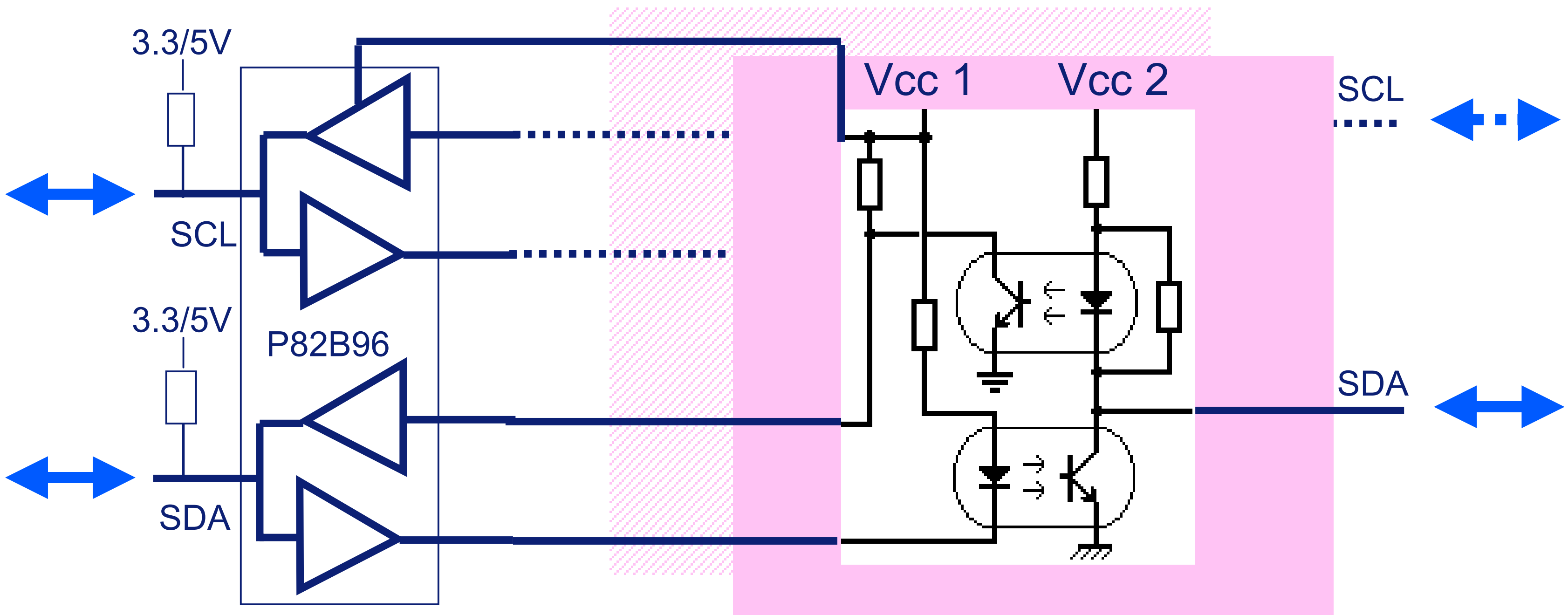
100 meters at 70kHz
NO LIMIT to the number of connected devices !

Re-combine to bi-directional I^2C

Convert the logic signal levels back to I^2C compatible

Hot Swap Protection

Changing I^2C bus signals for Opto-isolation



Bi-directional data streams

Special logic levels (I^2C compatible 5V)

I^2C currents (3mA)

Low cost Optos can be directly driven (10-30mA)

V_{CC} 1 = 2 to 12V

Higher current option, up to 30mA static sink

4N36 Optos for ~5kHz
6N137 for 100kHz

HCPL-060L for 400 kHz

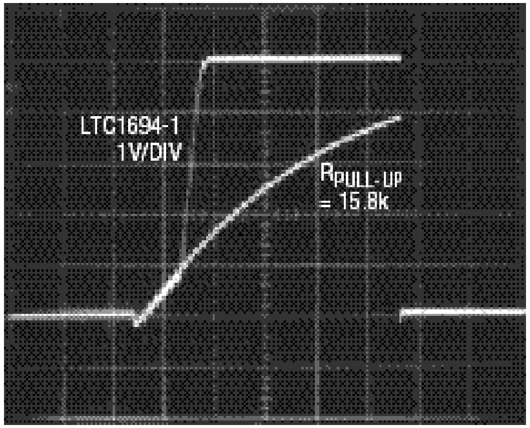
Re-combined to I^2C

I^2C compatible levels e.g. Vcc 2 = 5V

Controlling equipment on phone lines
AC Mains switches, lamp dimmers
Isolating medical equipment

Rise Time Accelerators

Comparison of SMBus Waveforms for the LTC1694-1 vs Resistor Pull-Up



V_{CC} = 5V
C_{LD} = 200pF
f_{SMBus} = 100kHz

1µs/DIV

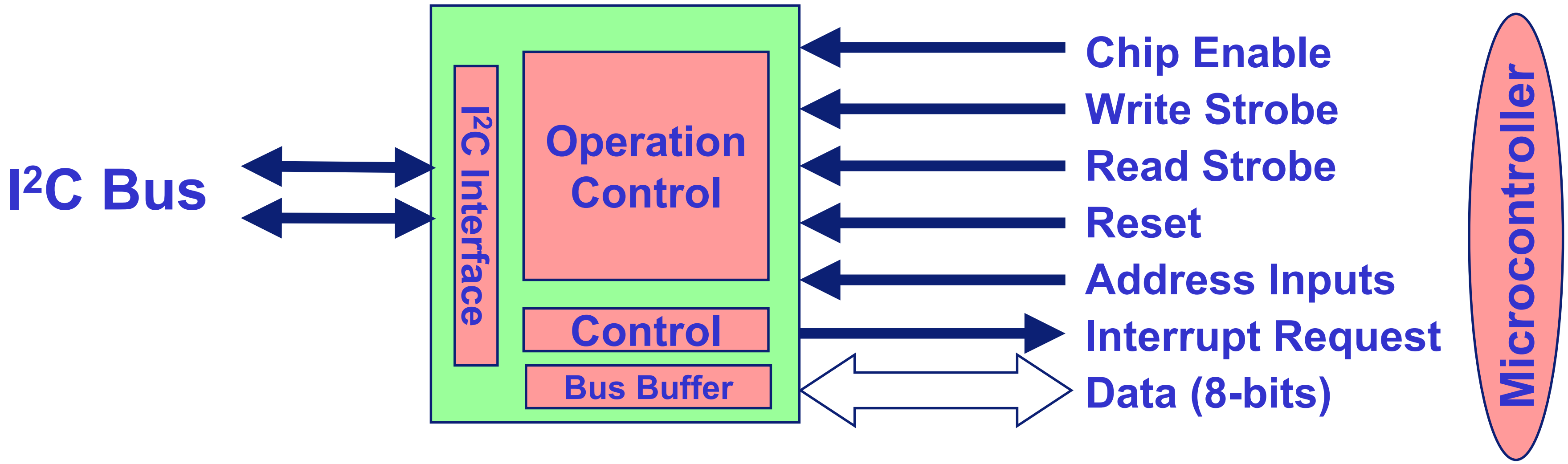
1694-1 TA02

The LTC®1694-1 is a dual SMBus active pull-up designed to enhance data transmission speed and reliability under all specified SMBus loading conditions. The LTC1694-1 is also compatible with the Philips I^{2}C Bus.

The LTC1694-1allows multiple device connections or a longer, more capacitive interconnect, without compromising slew rates or bus performance, by supplying a high pull-up current of 2.2 mA to slew the SMBus or I^{2}C lines during positive bus transitions

During negative transitions or steady DC levels, the LTC1694-1 sources zero current. External resistors, one on each bus line, trigger the LTC1694-1 during positive bus transitions and set the pull-down current level. These resistors determine the slew rate during negative bus transitions and the logic low DC level.

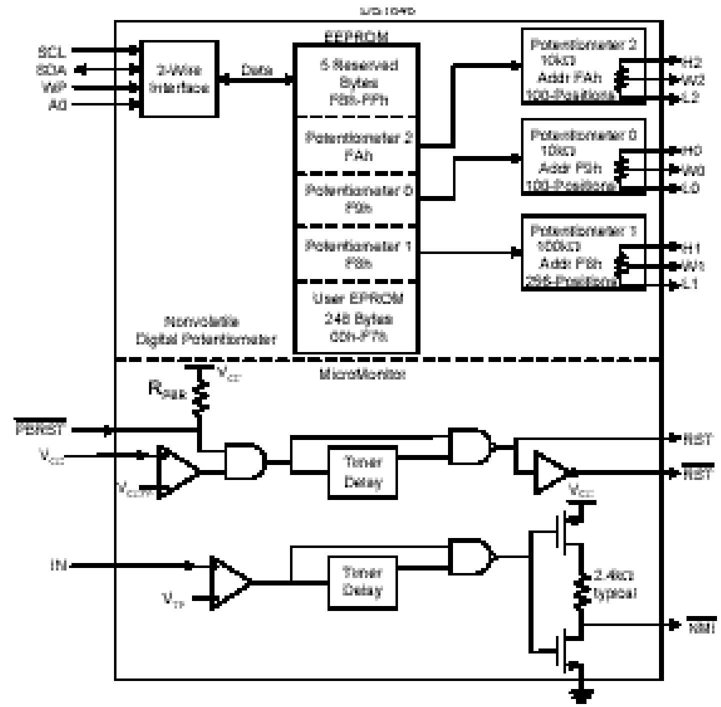
Parallel Bus to I²C Bus Controller



- Controls all the I²C bus specific sequences, protocol, arbitration and timing
- Serves as an interface between most standard parallel-bus microcontrollers/ microprocessors and the serial I²C bus.
- Allows the parallel bus system to communicate with the I²C bus

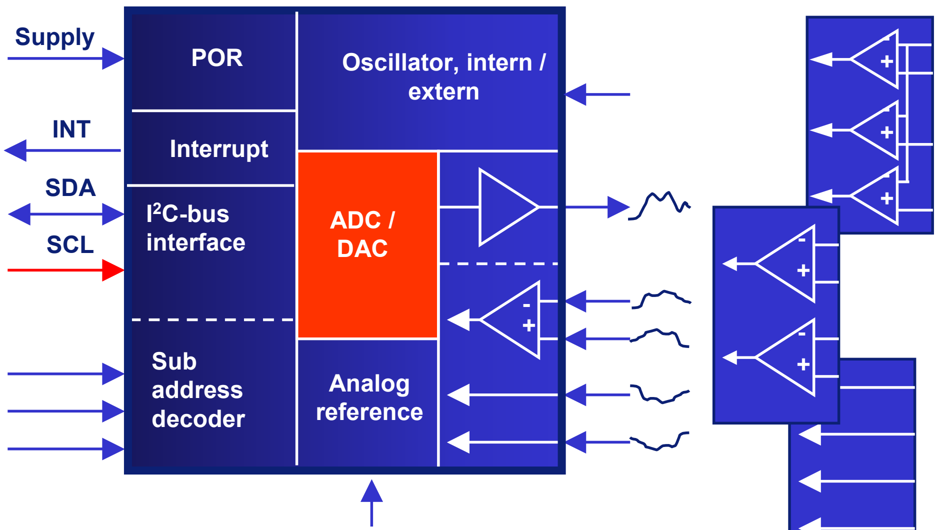
Digital Potentiometers

• DS1846 nonvolatile (NV) tri-potentiometer, memory, and MicroMonitor. The DS1846 is a highly integrated chip that combines three linear-taper potentiometers, 256 bytes of EEPROM memory, and a MicroMonitor. The part communicates over the industry-standard 2-wire interface and is available in a 20-pin TSSOP.



• The DS1846 is optimized for use in a variety of embedded systems where microprocessor supervisory, NV storage, and control of analog functions are required. Common applications include gigabit transceiver modules, portable instrumentation, PDAs, cell phones, and a variety of personal multimedia products.

Analog to Digital Converter



- 4 channel Analog to Digital
- 1 channel Digital to Analog

These devices translate between digital information communicated via the I²C bus and analog information measured by a voltage.

Analog to digital conversion is used for measurement of the size of a physical quantity (temperature, pressure …), proportional control or transformation of physical amplitudes into numerical values for calculation.

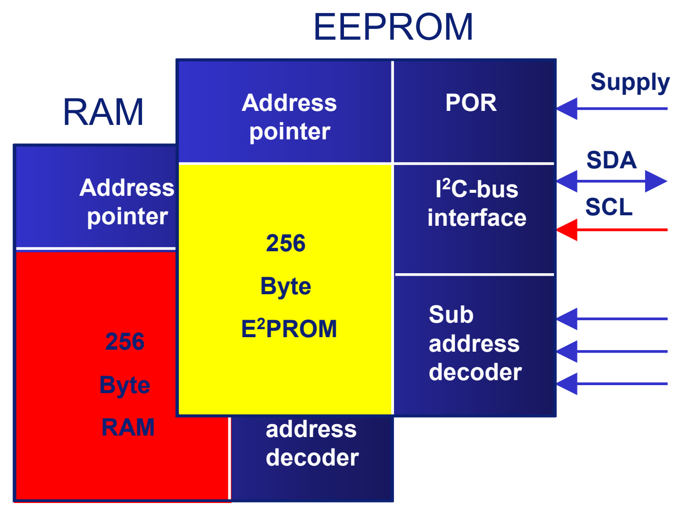
Digital to analog conversion is used for creation of particular control voltages to control DC motors or LCD contrast.

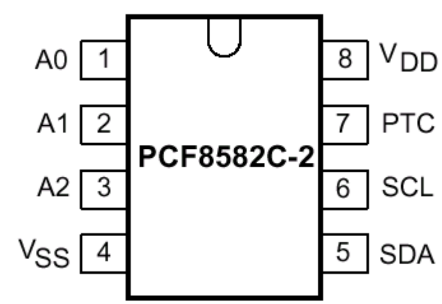
Blank

I²C Serial CMOS RAM/EEPROMs

Standard Sizes

128 x 8-byte (1 kbit)	**24C01**
256 x 8-byte (2 kbit)	**24C02**
512 x 8-byte (4 kbit)	**24C04**
1024 x 8-byte (8 kbit)	**24C08**
2048 x 8-byte (16 kbit)	**24C16**
4096 x 8-byte (32 kbit)	**24C32**
8192 x 8-byte (64 kbit)	**24C64**
16384 x 8-byte (128 kbit)	**24C128**
32768 x 8-byte (256 kbit)	**24C256**
65536 x 8-byte (512 kbit)	**24C512**

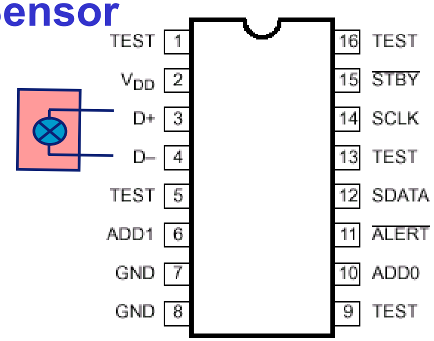




- I²C bus is used to read and write information to and from the memory
- Electrically Erasable Programmable Read Only Memory
    - 1,000,000 write cycles, unlimited read cycles
    - 10 year data retention

I^{2}C Hardware Monitors

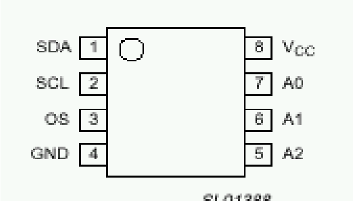
Remote Sensor



I^{2}C Temperature Monitor

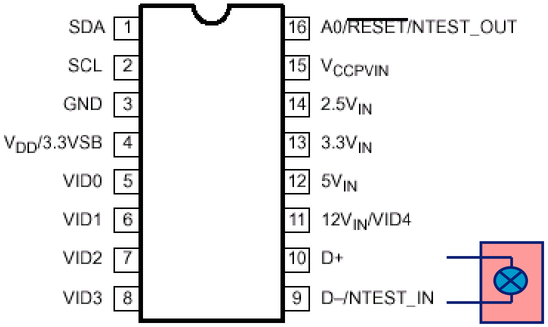
NE1617A

NE1618



Digital Temperature Sensor and Thermal Watchdog™

LM75A

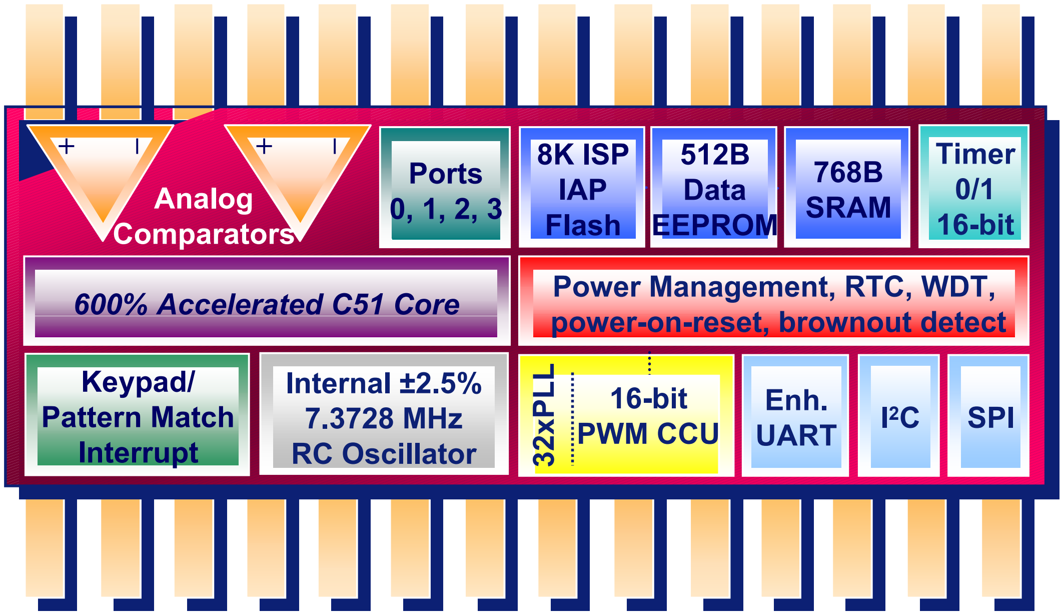


I^{2}C Temperature and Voltage Monitor(Heceta4)

NE1619

– Sense temperature and/or monitor voltage via I^{2}C
– Remote sensor can be internal to microprocessor

I²C Microcontroller



The master can be either a bus controller or µcontroller and provides the brains behind the I²C bus operation.

A bus controller adds I²C bus capability to a regular µcontroller without I²C, or to add more I²C ports to µcontrollers already equipped with an I²C port such as the:

P87LPC76x	100 kHz I²C
P89C55x	100 kHz I²C
P89C65x	100 kHz I²C
P89C66x	100 kHz I²C
P89LPC932	400 kHz I²C

Microcontrollers with Multiple Serial ports can convert from:
I²C to UART/RS232 – LPC76x, 89C66x and 89LPC9xx
I²C to SPI - P87C51MX and 89LPC9xx family
I²C to CAN - 8 bit P87C591 and 16 bit PXA-C37

I^{2}C Patent and Legal Information

I^2C Patent Information

- The I^2C bus is protected by patents held by Philips. Licensed IC manufacturers that sell devices incorporating the technology already have secured the rights to use these devices, relieving the burden from the purchaser.

- A license is required for implementing an I^2C interface on a chip (IC, ASIC, FPGA, etc). It is Philips's position that all chips that can talk to the I^2C bus must be licensed. It doesn’t matter how this interface is implemented. The licensed manufacturer may use its own know how, purchased IP cores, or whatever.

- This also applies to FPGAs. However, since the FPGAs are programmed by the user, the user is considered a company that builds an I^2C-IC and would need to obtain the license from Philips.

- Apply for a license or text of the Philips I^2C Standard License Agreement
  - US and Canadian companies: contact Mr. Piotrowski (pc.mb.svl@philips.com)
  - All other companies: contact Mr. Hesselmann (ps.mb.svl@philips.com)

Questions And Answers

Philips Semiconductors

Specialty Logic Product Line

Booth 836

Download AN10126-01 I^2C Manual for speaker notes for this presentation