
# Interface Solutions

## $I^2C$-bus

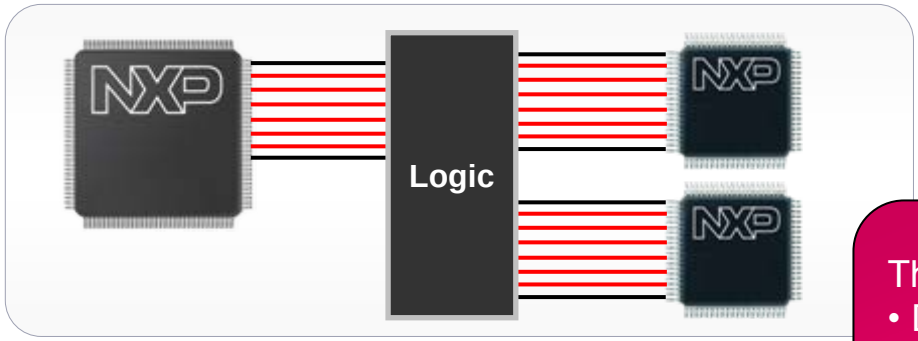
30 July, 2008

# I²C - Contents

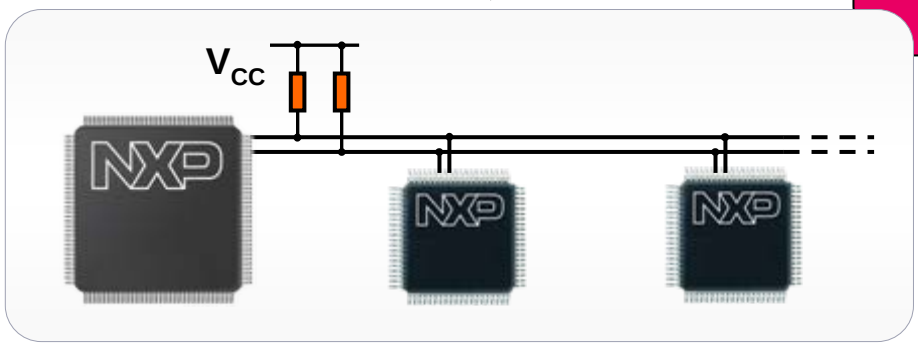
- I²C protocol
- I²C functions
- I²C bus enablers
- I²C demo boards
- I²C support information

# I²C - Protocol

## IIC - Inter-Integrated Circuit




I²C-bus



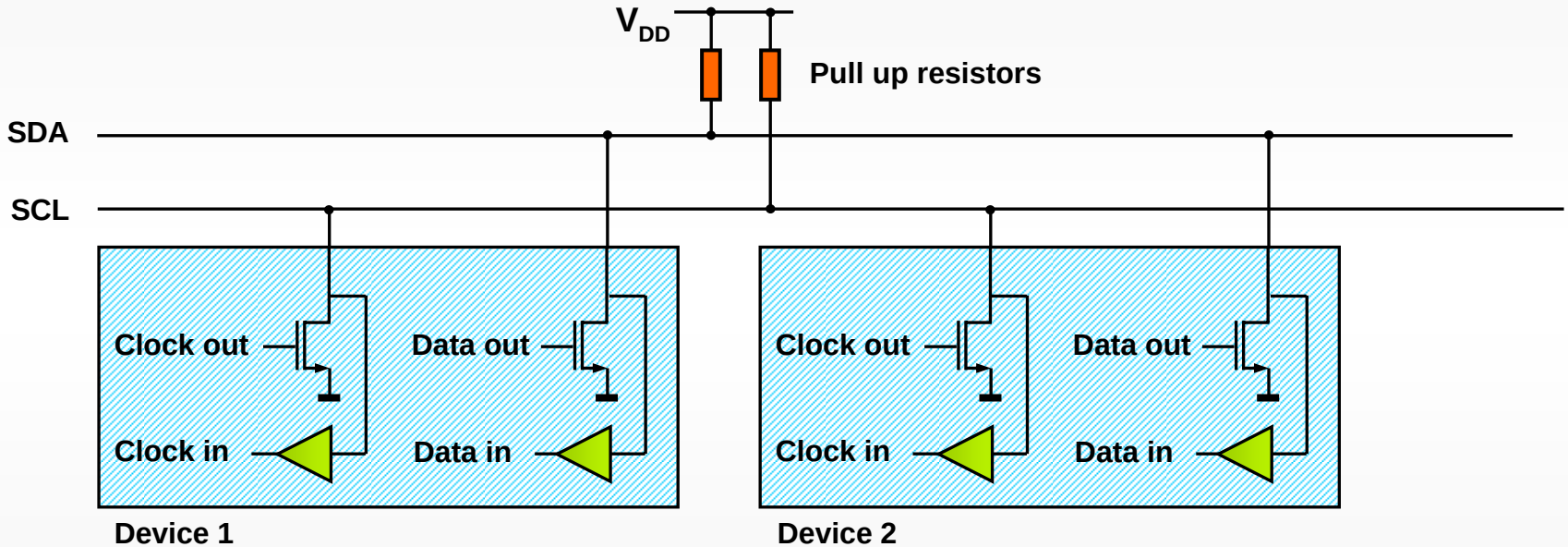

This means:
- Decreased number of wires (reduced PCB area)
- Reduced number of chip pins
- Remove glue logic
- Clip many devices on to the bus
- Modular design: Time-to-Market

**Invented by NXP!**
**(Philips Semiconductors)**

- I²C-bus developed in the late 1970's for Philips consumer products (e.g. TVs)
- Worldwide industry standard and used by all major IC manufacturers

# I²C - Protocol

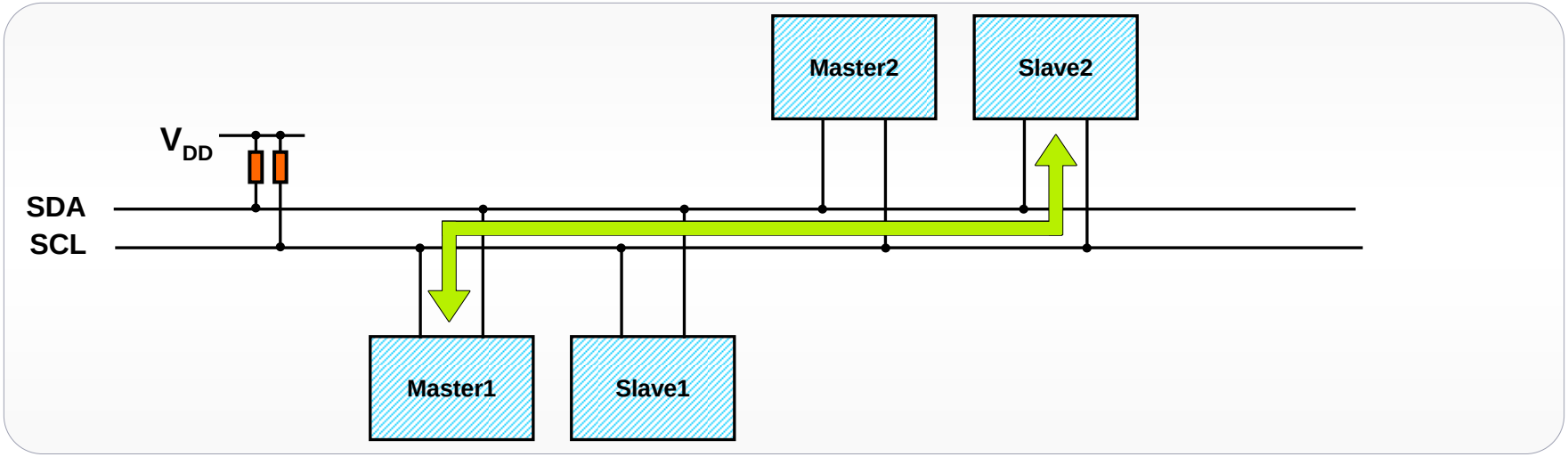
## Hardware architecture




- 2 wire bus:
  - *SDA*: Serial *Data* Line
  - *SCL*: Serial *Clock* Line
- Open-drain or open-collector output stages: wired-AND function

# I²C - Protocol

## Hardware architecture (2)

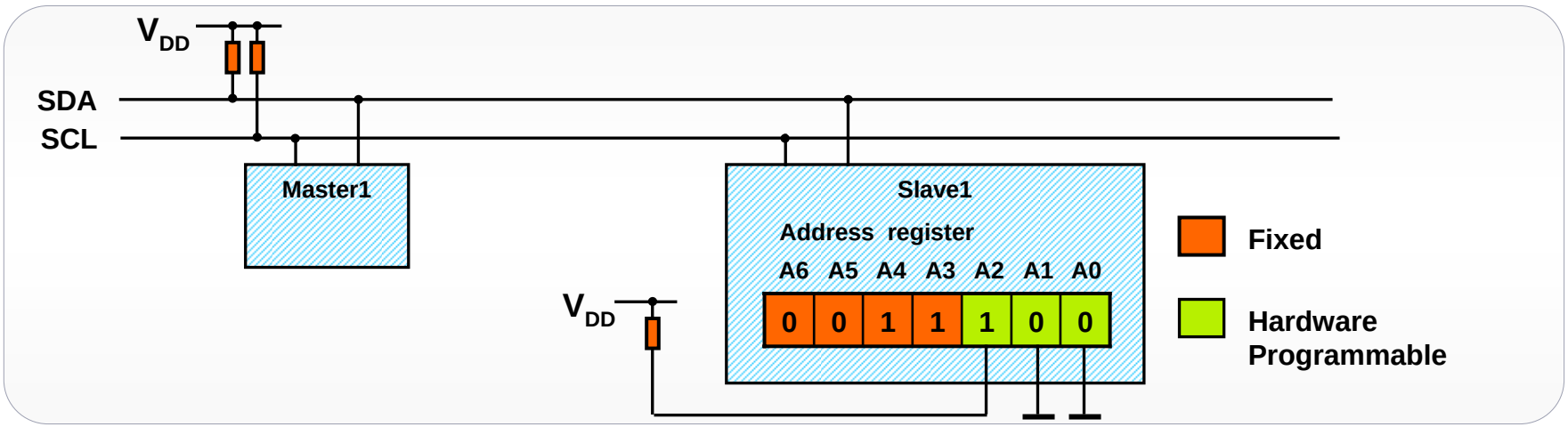



- Multiple master
- Multiple slave
- Bi-directional
  - Master-transmitter
  - Master-receiver
  - Slave-transmitter
  - Slave-receiver
- Data collision is taken care off

# I²C - Protocol

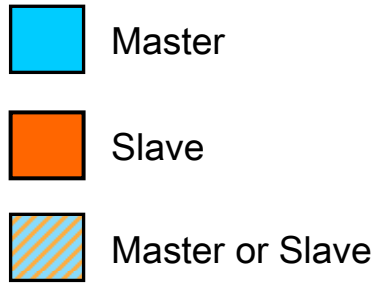
## Addressing / device selection

- Each device is addressed individually by software
- New devices or functions can be easily “clipped" on to an existing bus!
- 112 different addresses max with the 7-bit format (others reserved); additional 1024 with 10-bit format
- Address allocation coordinated by the I²C-bus committee
- Programmable pins means that several of the same devices can share the same bus
- Unique address per device: fully fixed or with a programmable part through hardware pin's)
- 10-bit format use a 2 byte message: 1111 $0A_9A_8$R/W + $A_7A_6A_5A_4A_3A_2A_1A_0$



# I²C - Protocol

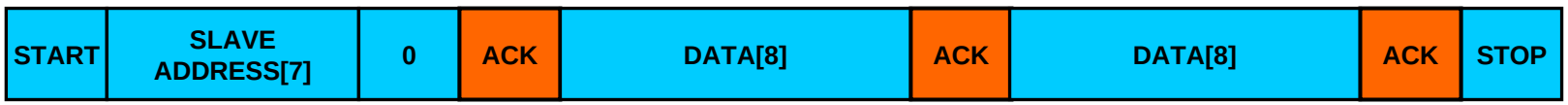
## Communication




- Communication must start with: START condition
- Start bit is always followed by slave address
- Slave address is followed by a READ or NOT-WRITE bit
- The receiving device (either master or slave) must send an ACKNOWLEDGE bit
- Communication must start with: STOP condition


- Example:

  Transmit (0 = Write)

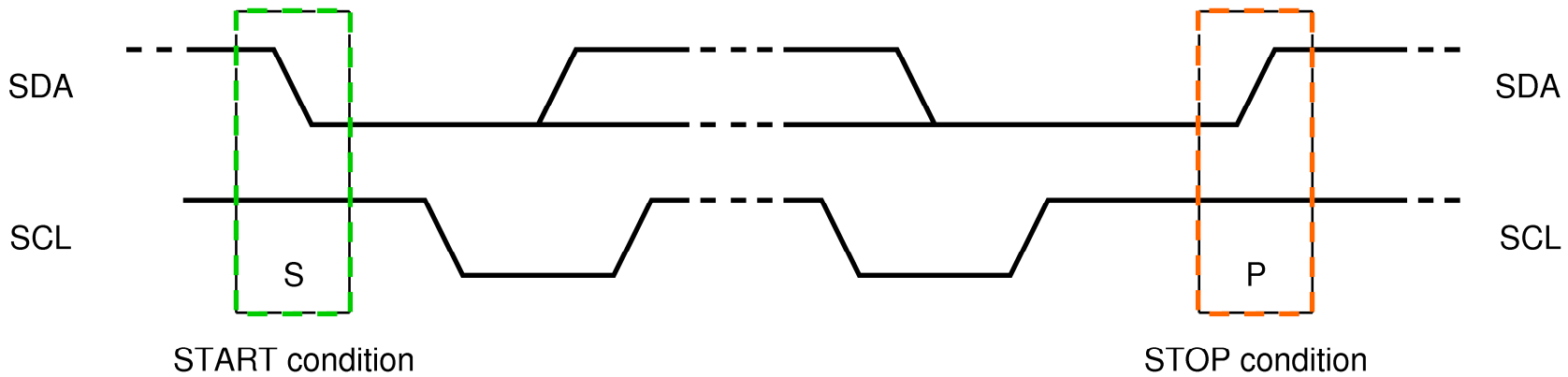



  Receive (1 = Read)

# I²C - Protocol

## START & STOP conditions

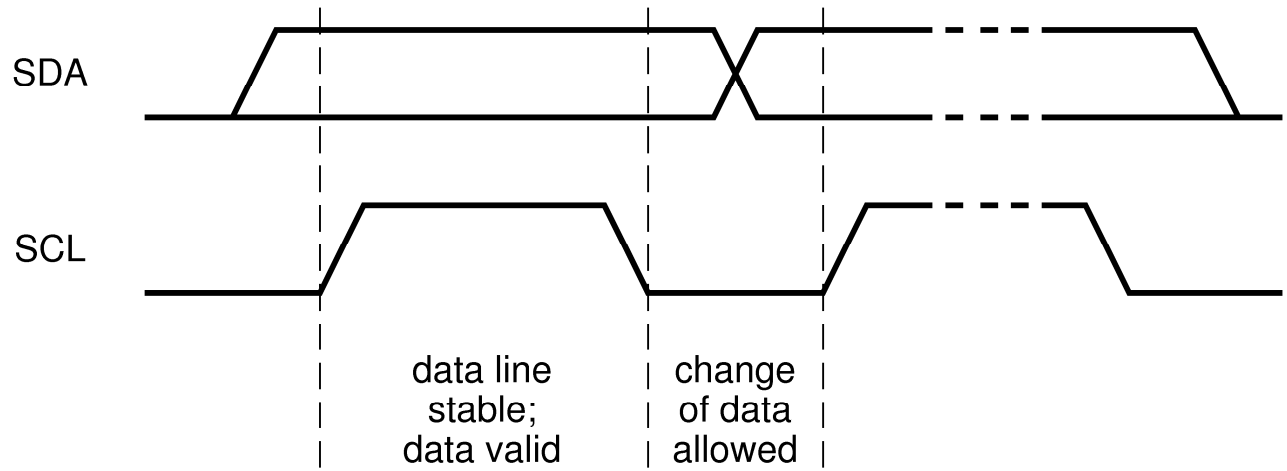
- Start condition - a HIGH to LOW transition on the SDA line while SCL is HIGH
- Stop condition - a LOW to HIGH transition on the SDA line while SCL is HIGH



# I²C - Protocol

## Bit transfer

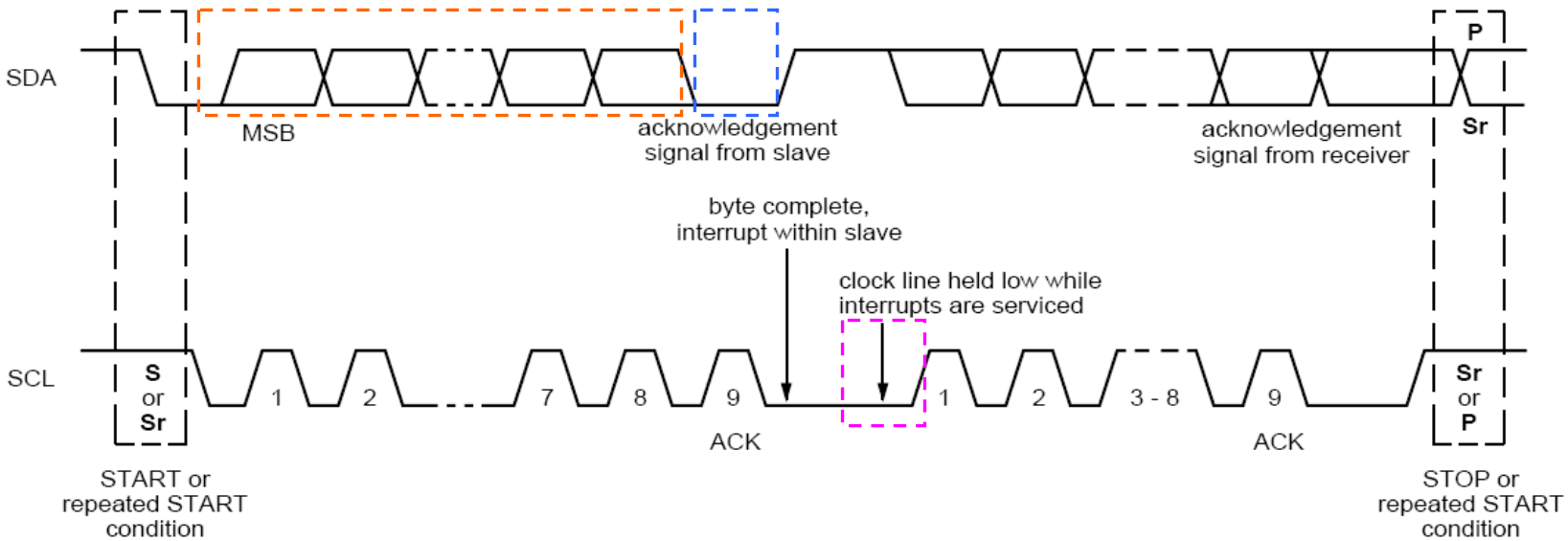
- During data transfer, SDA must be stable when SCL is High



# I²C - Protocol

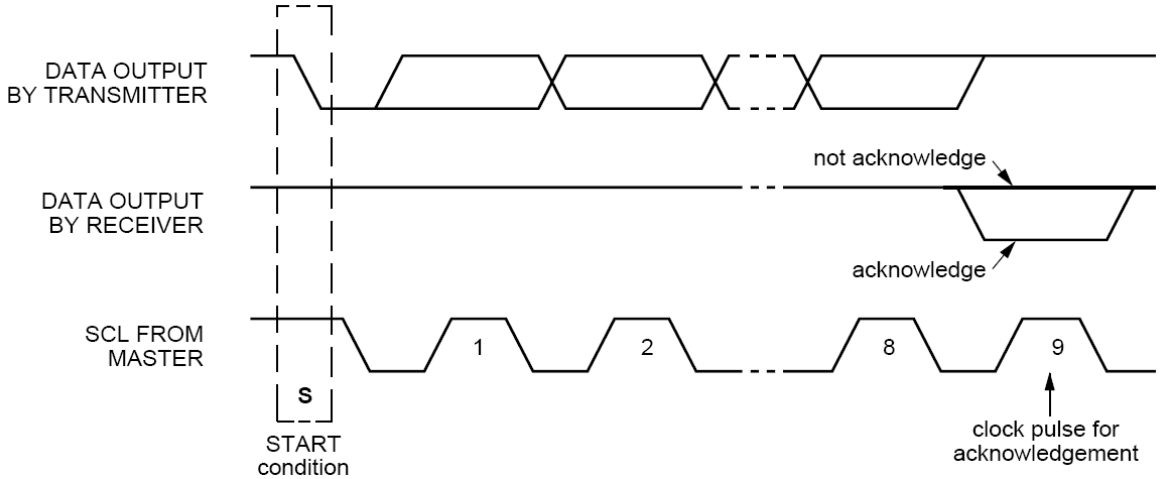
## Data transfer

- Each byte has to be followed by an acknowledge bit
- Number of data bytes transmitted per transfer is unrestricted
- If a slave can't receive or transmit another complete byte of data, it can hold the clock line SCL LOW (clock stretching) to force the master into a wait state



# I²C - Protocol

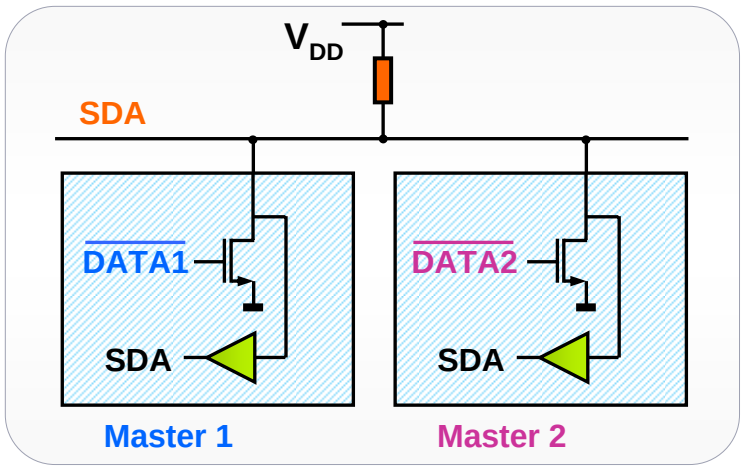
## Acknowledge / NOT-Acknowledge




- I²C specification: Data transfer with acknowledge is obligatory. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse.
- Scenarios with a NOT-acknowledge (NACK) (SDA staying HIGH):
  1. A receiver with the address is not present in the I²C bus.
  2. The receiver is performing real-time tasks and it cannot process the received I²C information.
  3. The receiver is the master and wants to take control of SDA line again in order to generate a STOP command. The slave transmitter MUST then release the SDA line when it sees the NACK so the master can send the STOP command.

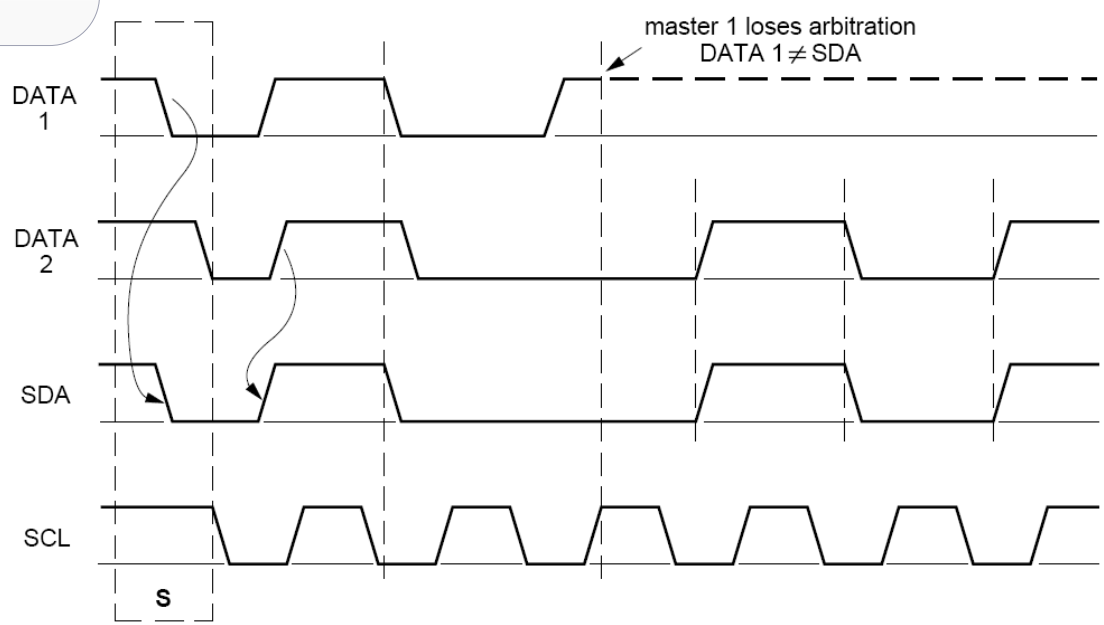
# I²C - Protocol

## Arbitration procedure



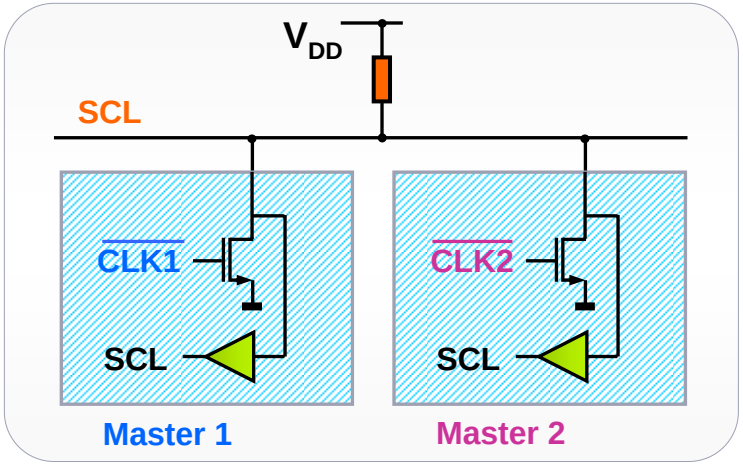

- Two or more masters may generate a START condition at the same time
- Arbitration is done on SDA while SCL is HIGH - Slaves are not involved

Summary: The master that first sends a “1” while the other sends a “0” loses control (arbitration)

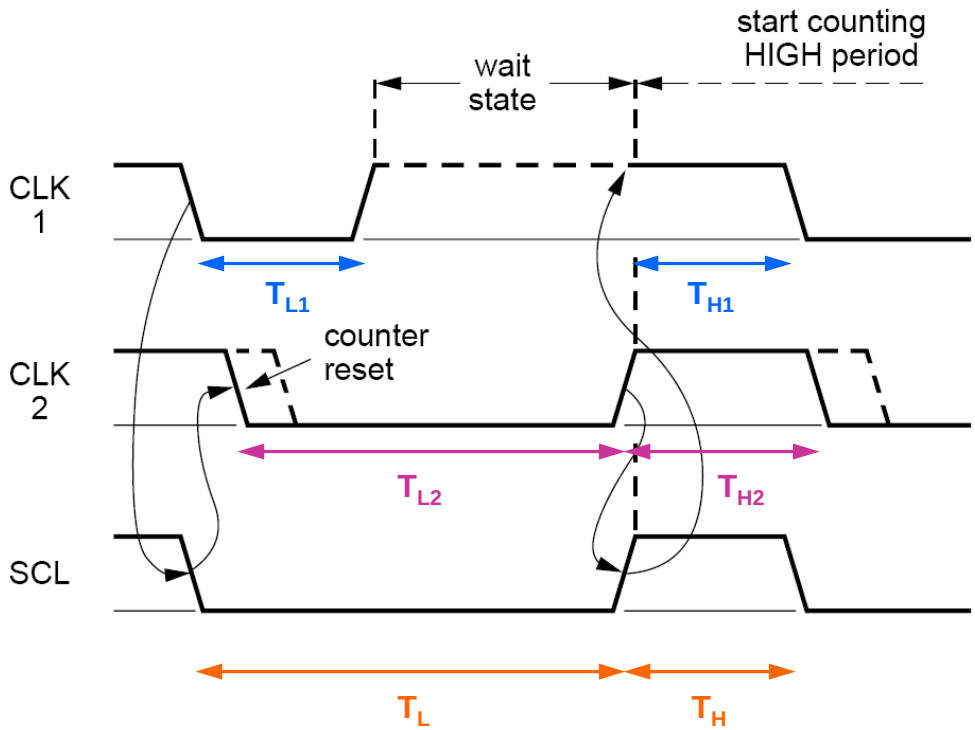


# I²C - Protocol

## Clock synchronization during the arbitration procedure




Internal counters of masters count the LOW and HIGH times ($T_{L1}$, $T_{H1}$) and ($T_{L2}$, $T_{H2}$)




Wired-AND SCL connection:
$T_L$= longest $T_L$= max ($T_{L1}$, $T_{L2}$ ,$T_{Ln}$)
$T_H$= shortest $T_H$= min ($T_{H1}$, $T_{H2}$,$T_{Hn}$)

# I²C - Protocol

## Modes

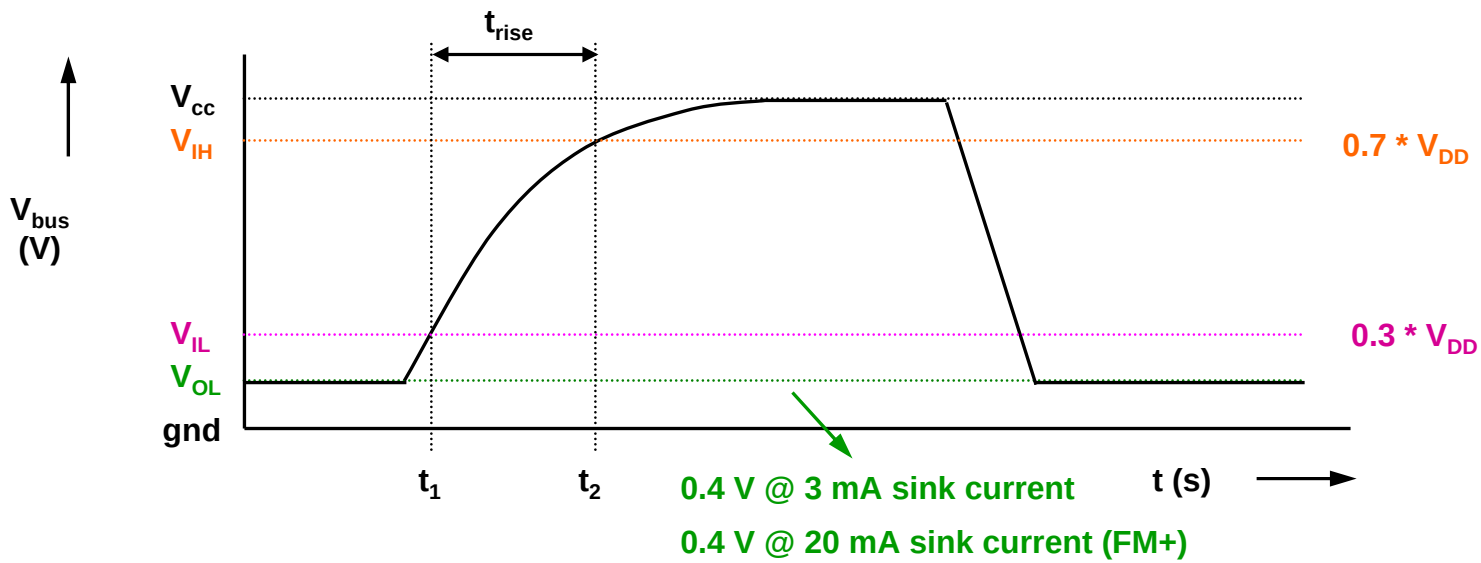

| | Standard Mode | Fast Mode | Fast Mode Plus (FM+) | High Speed Mode | |
|---|---|---|---|---|---|
| Bitrate (kBit/s) | 0 – 100 | 0 – 400 | 0 – 1000 | 0 – 1700 | 0 – 3400 |
| Address (bits) | 7 (10) | 7 (10) | 7 (10) | 7 (10) | 7 (10) |
| Capacitive Bus Load (pF) | 400 | 400 | 550 | 400 | 100 |
| Sink current (mA) | 3 | 3 | 20 | 3 | 3 |

- Fast mode Plus (FM+):
  - Increased bandwidth
  - Increased transmission distance (at reduced bandwidth: >> 550 pF bus load)

# I²C - Protocol

## Modes: Electrical specification

| | Standard Mode | Fast Mode | Fast Mode Plus (FM+) | High Speed Mode | |
|---|---|---|---|---|---|
| Bitrate (kBit/s) | 0 – 100 | 0 – 400 | 0 – 1000 | 0 – 1700 | 0 – 3400 |
| Address (bits) | 7 (10) | 7 (10) | 7 (10) | 7 (10) | 7 (10) |
| Capacitive Bus Load (pF) | 400 | 400 | 4000 | 400 | 100 |
| Sink current (mA) | 3 | 3 | 20 | 3 | 3 |
| Trise: Rise time (ns) | 1000 | 300 | 120 | 160 | 80 |



# I²C - Protocol

## Electrical: $V_{DD}$ / How to calculate the pull-up resistor values

- $V_{DD}$voltage can be chosen freely
- Pull-up resistor value:
  - Minimum resistor value:
    - Determined by the I²C spec limit of 3 mA sinking current
    - $R = (V_{DDmax} – V_{olmax})/ 0.003A$
    - Example: using a $V_{DD}$ = 5 ± 0.5 V: $R_{pull\text{-}up}$ = (5.5 V – 0.4 V) / 0.003 A = 1.7 kΩ
  - Maximum resistor value:
    - Determined by the I²C-bus rise time requirements:
      $V(t_1) = 0.3 * V_{DD} = V_{DD} (1 – 1/e^{t1/RC})$; then $t_1 = 0.3566749 * RC$
      $V(t_2) = 0.7 * V_{DD} = V_{DD} (1 – 1/e^{t2/RC})$; then $t_2 = 1.2039729 * RC$
      $t = t_2 – t_1 = 0.8472979 * RC$
    - For standard-mode I²C-bus: $t_{rise}$ = 1000 ns (1 µs)
      so RC = 1180.2 ns
    - Example: at a bus load of 400 pF: $R_{max}$= 2.95 kΩ
    - For Fast-Mode: I²C-bus rise time = 300 ns @ 400 pF: $R_{max}$= 885 Ω

# 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):<br>- Must be stable when SCL is HIGH<br>- Can change only when SCL is LOW<br>- Number of bytes transmitted is unrestricted |
| ACKNOWLEDGE | - Done on each 9th clock pulse during the HIGH period<br>- The transmitter releases the bus - SDA goes HIGH<br>- The receiver pulls DOWN the bus line - SDA goes LOW |
| CLOCK | - Generated by the Master(s)<br>- Maximum speed: (100, 400, 1000, 3400 kHz) but NO min<br>- A receiver can hold SCL low when performing another function (transmitter in a Wait state)<br>- A master can slow down the clock for slow devices |
| ARBITRATION | - Master can start a transfer only if the bus is free<br>- Several masters can start a transfer at the same time<br>- Arbitration is done on SDA line<br>- Master that lost the arbitration must stop sending data |

# I²C - Protocol

## Demoboard I2C 2005-1: software & hardware setup

Proceed as follows if WinI2CUSB Lite is not installed on your computer:

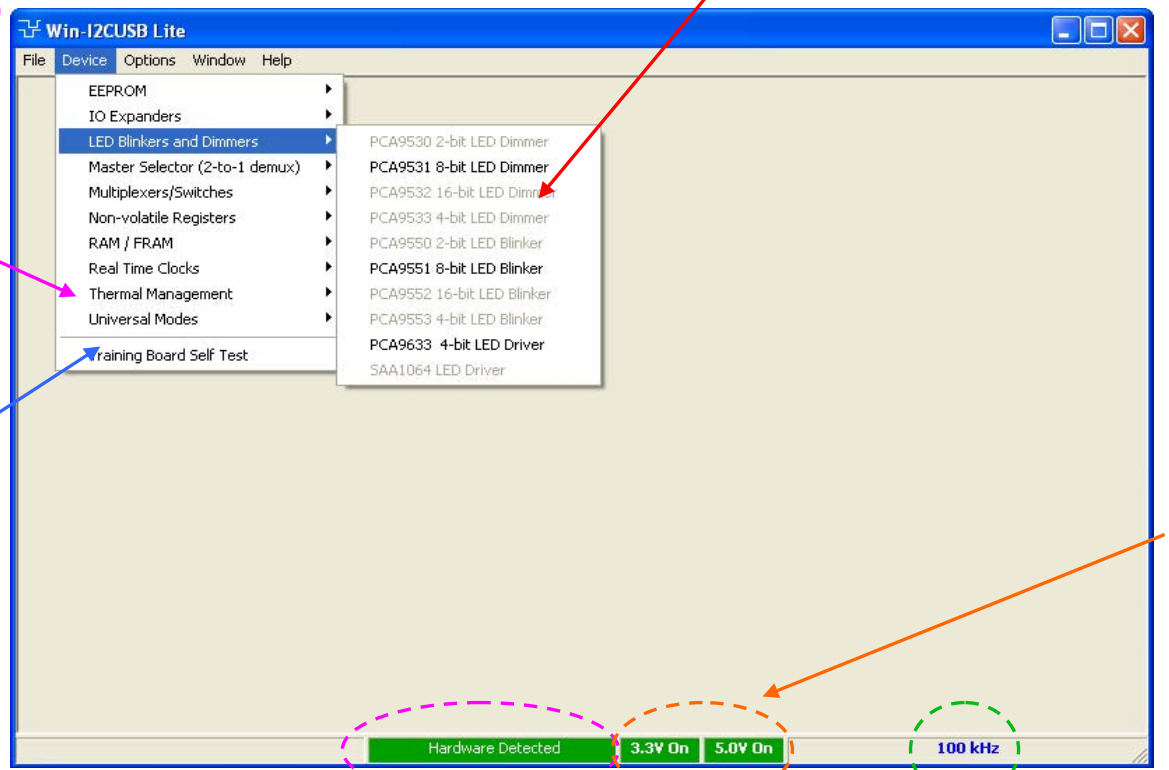
1. Run *Win-I2CUSB Lite Install* from the *software.win-i2cusb.lite.zip* file
2. Replace the *Win-I2CUSB Lite Installation.exe* executable file by the latest version (if necessary). Default installation path is: *C:\Program Files\The Boardshop\Win-I2CUSB Lite*
3. Plug the I²C Demonstration Board I2C 2005-1 into a USB port
4. Install the USB Driver. Default installation path is: *C:\Program Files\The Boardshop\Win-I2CUSB Lite\USBDriver* The hardware should now be ready to use.
5. Run *Win-I2CUSB Lite.exe*

# I²C - Protocol

## Demoboard I2C 2005-1: Win-I2CUSB LiteGUI Overview

Device selection: only devices in the board can be selected

Expert Mode allows to write customized I²C commands limited to 8 commands with 8 bytes of data for each




Self test feature: allows board diagnostic to be performed

Control 3.3 V and 5.0 V power supplies. Double-click enables/disables the selected power supply. Also available under the Options menu

Hardware detection
Green = Hardware detected
Red = Hardware not detected

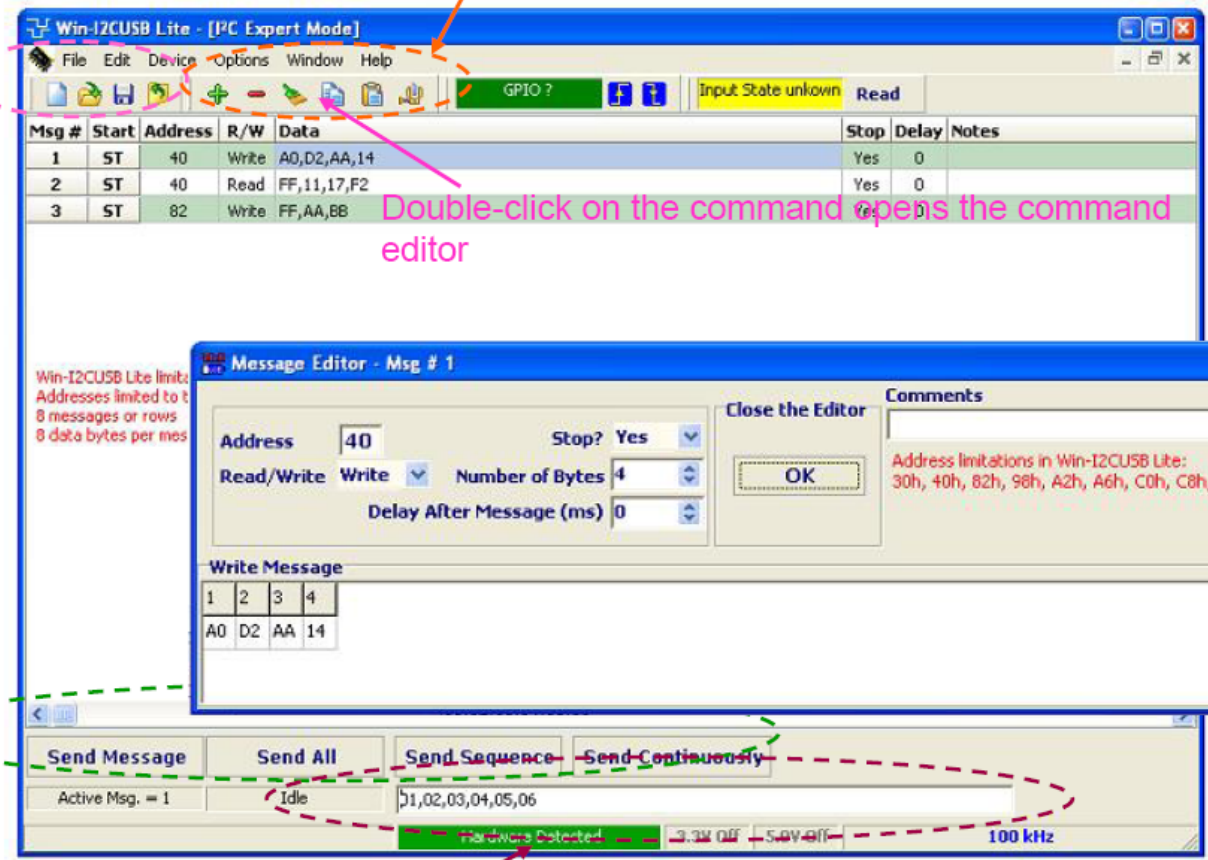
I²C Clock frequency Double-clicking allows changing the frequency.
Also available under the Options menu

# I²C - Protocol

## Demoboard I2C 2005-1: Expert Mode

File management:
New, Open, Save

I²C Message management:
Remove, Add, Copy, Paste, Compress




Up to 8 lines of code

Each line with 8 data bytes max

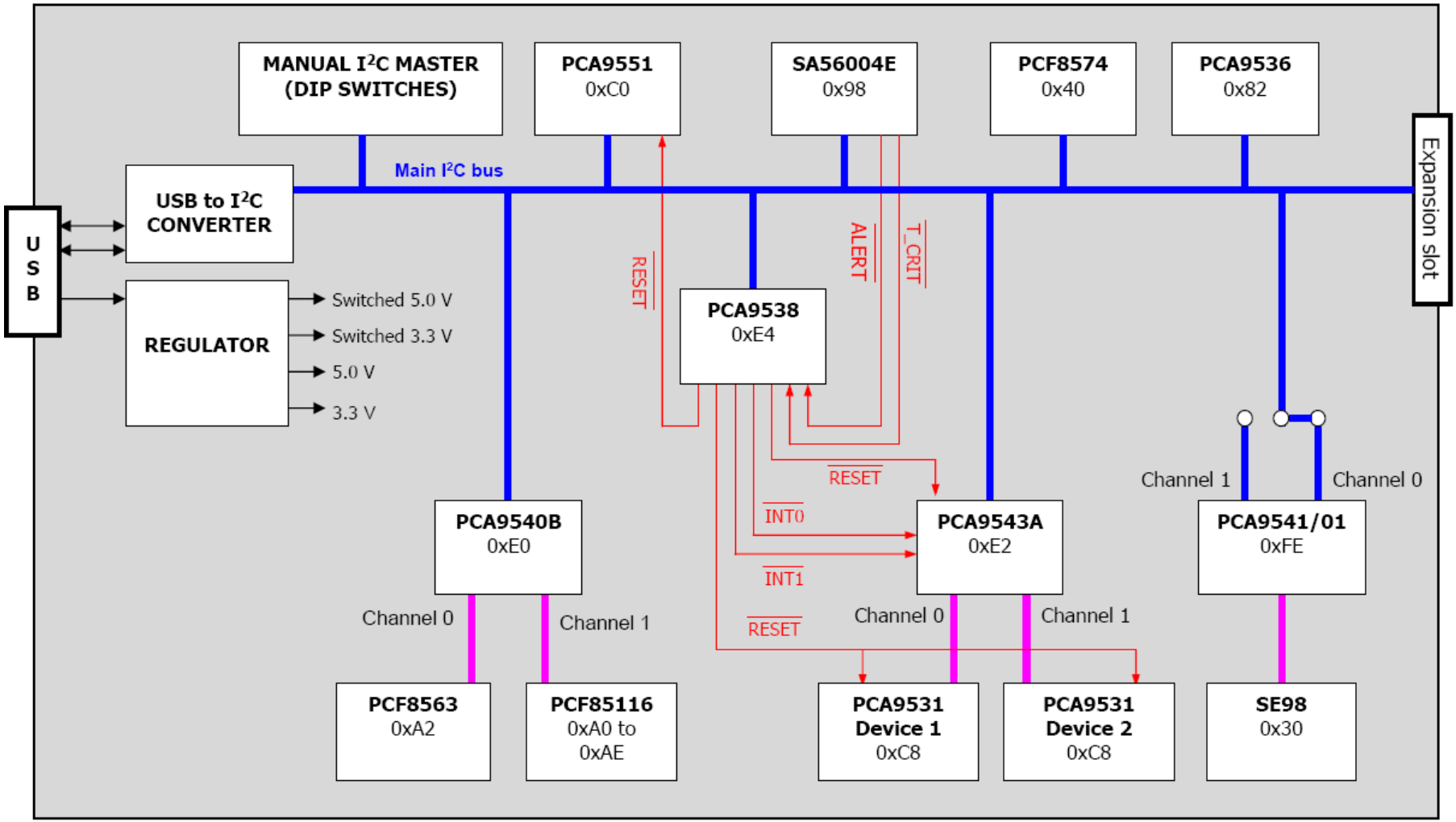
Double-click on the command opens the command editor

I²C Command sequencer

Sequence management
Double click opens the sequence editor

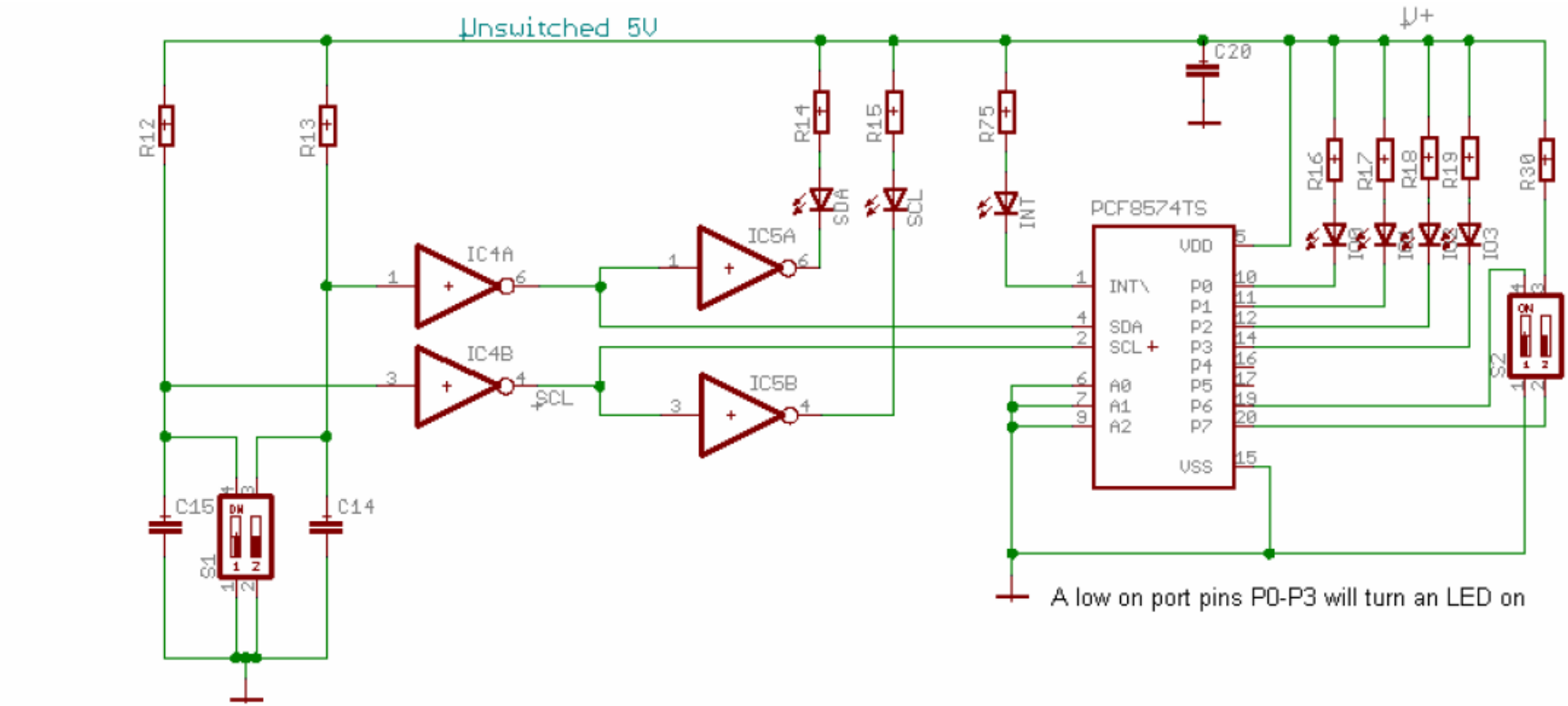
# I²C - Protocol

## Demoboard I2C 2005-1: Demonstration Board Block Diagram



# I²C - Protocol

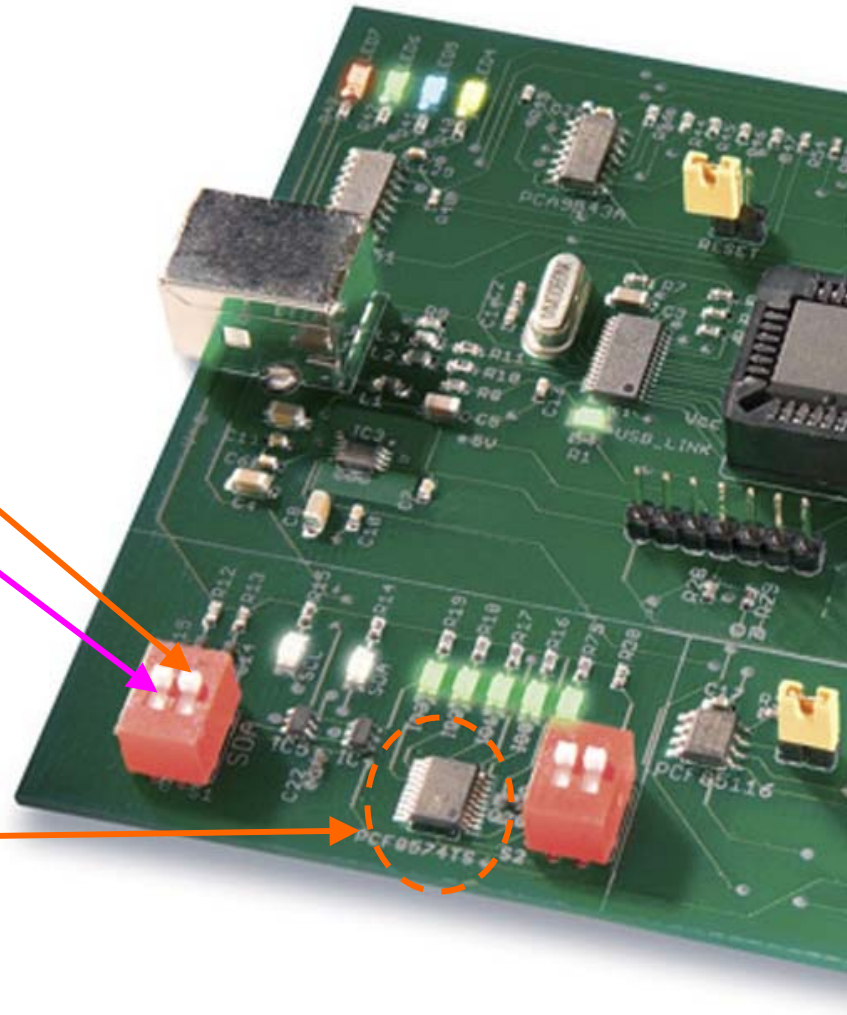
## Demoboard I2C 2005-1: PCF8574 schematic



# I²C - Protocol

## Demoboard I2C 2005-1: PCF8574 on the board

1. Switch S1: SDA
2. Switch S1: SCL



# I²C - Protocol

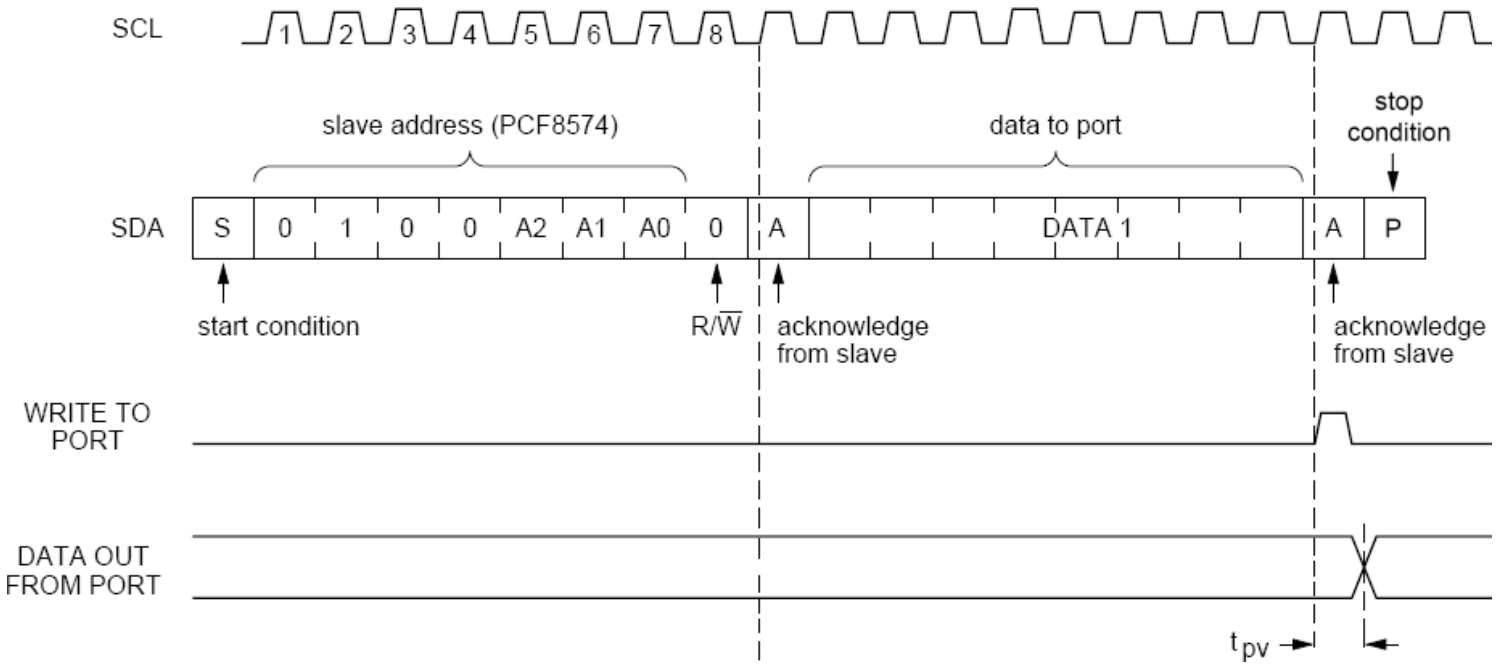
## Demoboard I2C 2005-1: Exercise: Use switch to generate an I²C message

- Objective: verify understanding of the I²C bus protocol
- Use switch S1 to generate an I²C message to the PCF8574:
  - The PCF8574 address is 0x40
  - The PCF8574 requires one data byte
  - The data byte should be 0xFA to turn on two of the LEDs

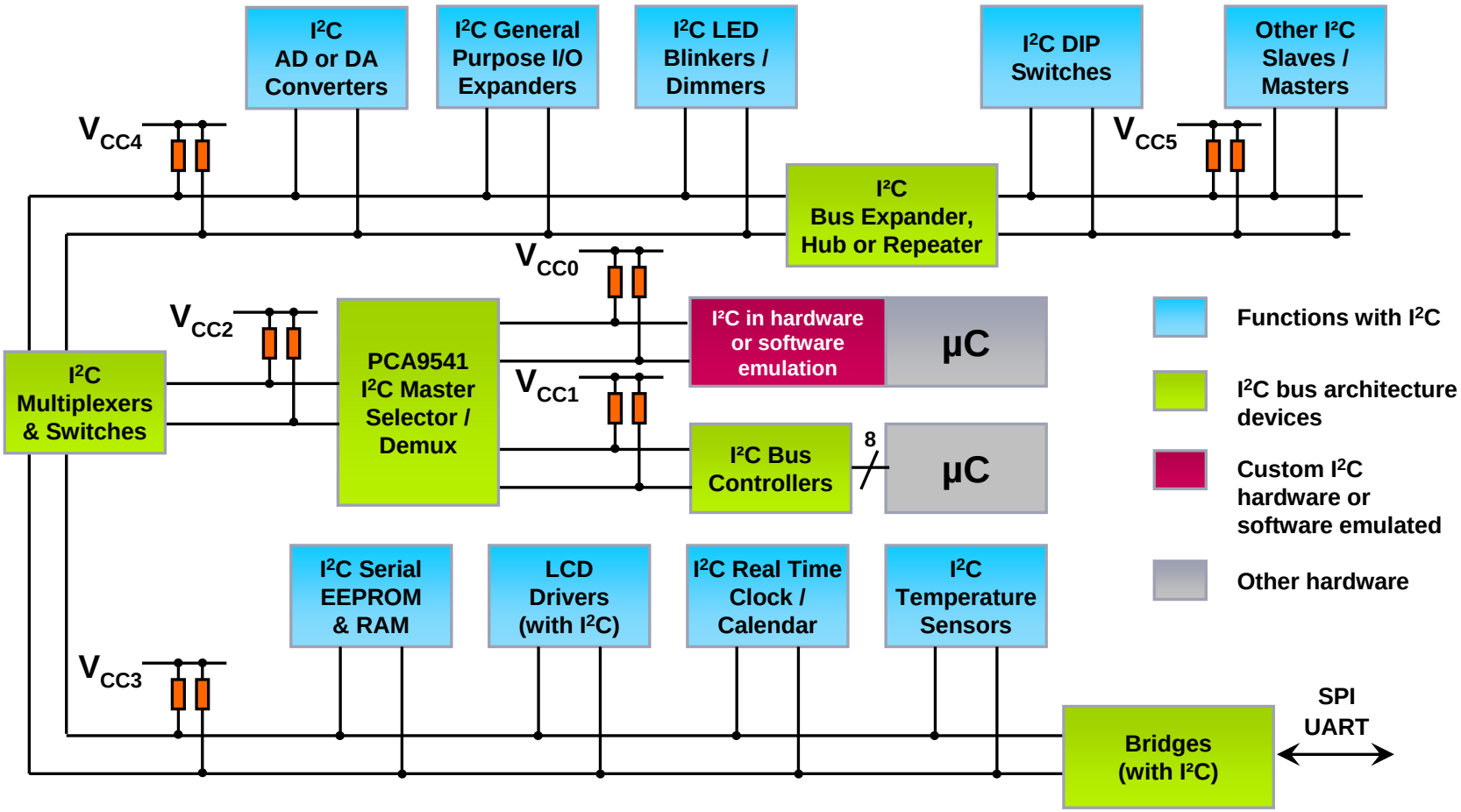
# I²C - Protocol

## Demoboard I2C 2005-1: Exercise: solution

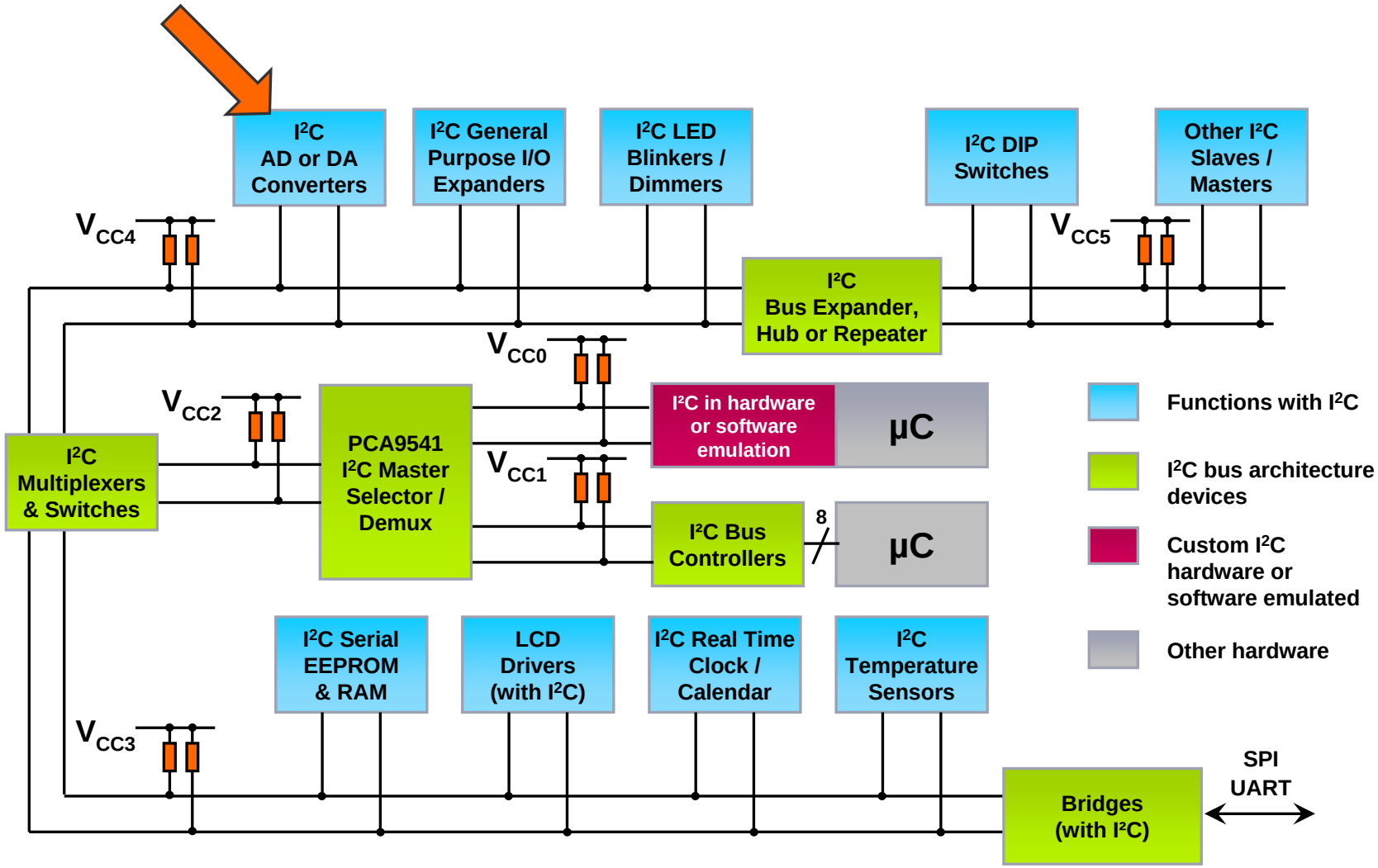
- The following sequence should be followed:
  - Send Start condition
  - Send PCF8574 address with LSB= ‘0’ for Write: 0100 0000b
  - Send data byte 0xFA: 1111 1010b
  - Send Stop condition
  - Don’t forget the acknowledge clock pulses



# I²C - Building blocks



# I²C - AD / DA Converters



# I²C - AD / DA Converters

## Application

8. Read out of sensors

9. Audio distribution

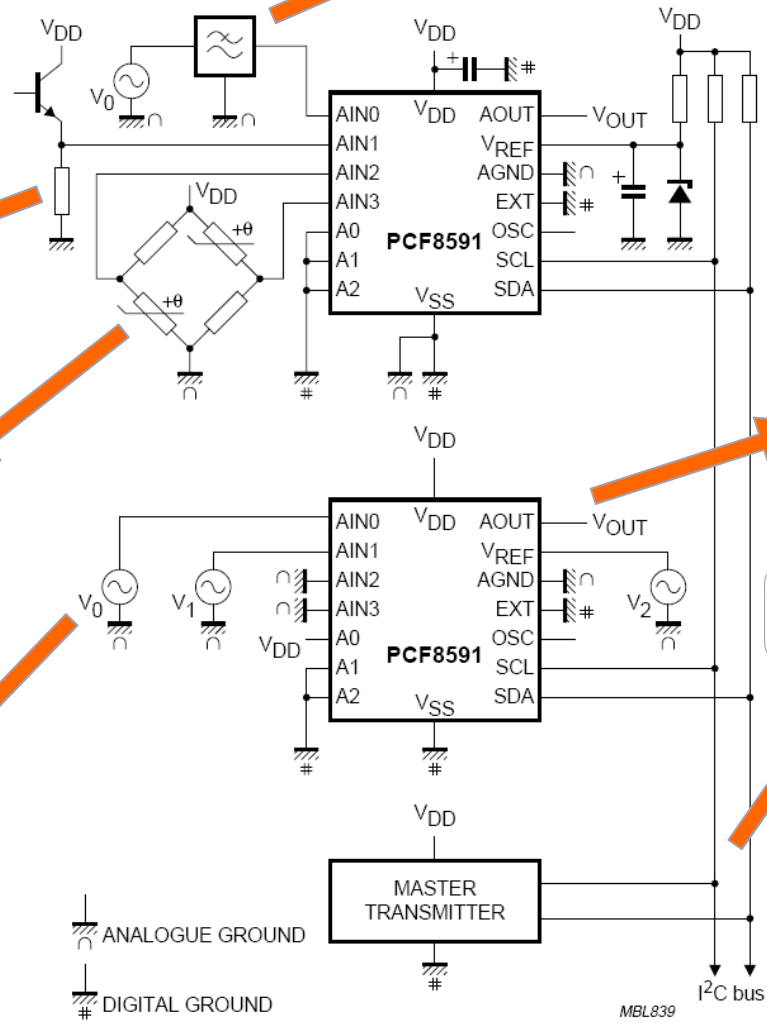
1. Read out of grounded voltages

2. Read out of Wheatstone-bridges: measure resistance very accurately

6. Use DAC output and AD input to implement fast control loop

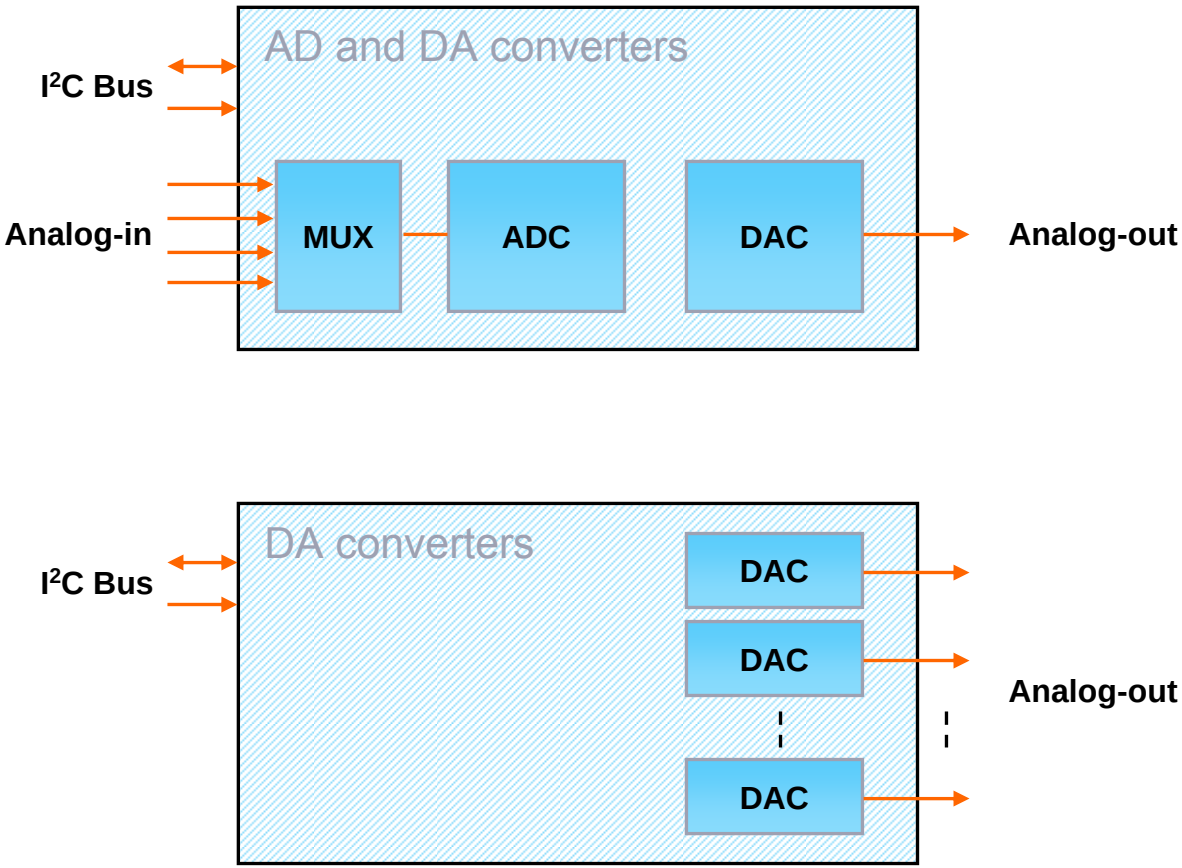
5. Read out remotely

4. Battery monitoring / Floating voltages



# I²C - AD / DA Converters

## Product overview



# I²C - AD / DA Converters

## AD and DA converters


### PCF8591

- 8-bit resolution
- Simply add the capability to:
  - 4 analog inputs
  - One analog output
- Sampling rate: 11 kHz
- Can support a large variety of input configurations

### PCA9691

- See PCF8591 +
- Fast Mode Plus (FM+) device:
  - High sampling rate: 110 kHz
  - Allows audio signal distribution
  - Fast control loops without occupying the bus continuously

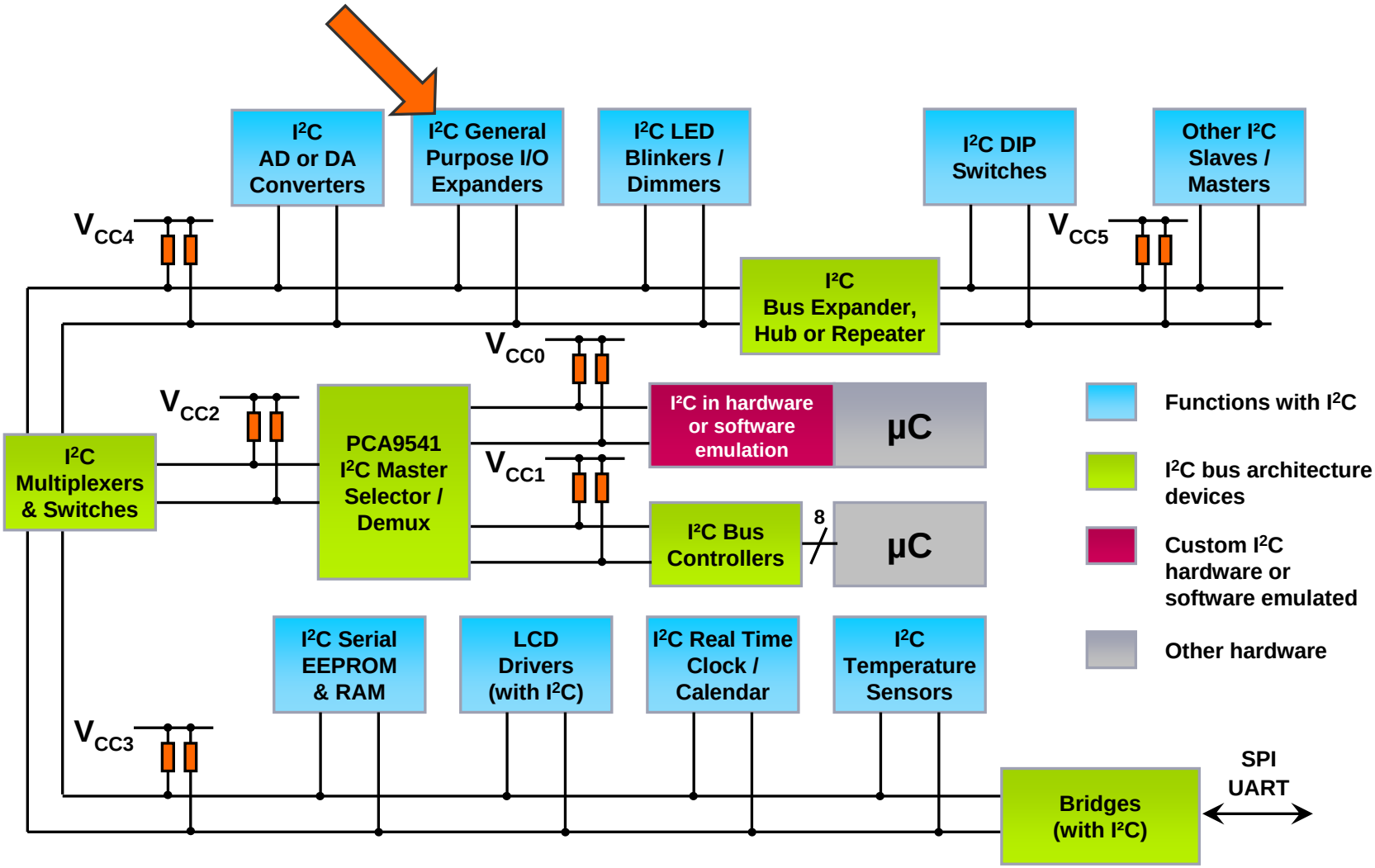
# I²C - AD / DA Converters

## DA converter

TDA8444

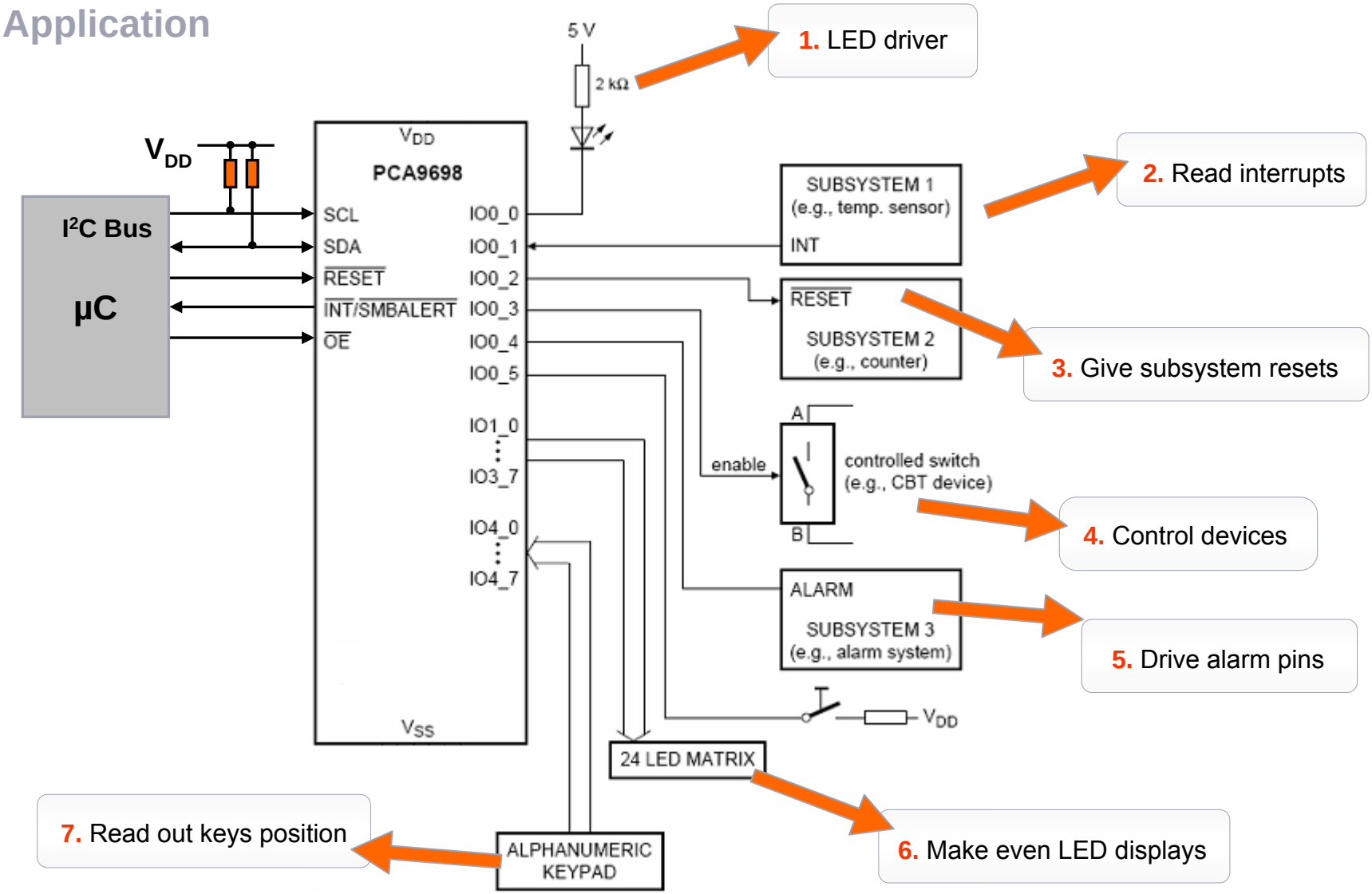
- Eight DACs with 6-bit resolution
- Adjustable common output swing
- Push-pull outputs
- Outputs short-circuit protected
- Three programmable slave address bits
- Large supply voltage range
- Low temperature coefficient

# I²C - GPIOs: General Purpose I/O Expanders



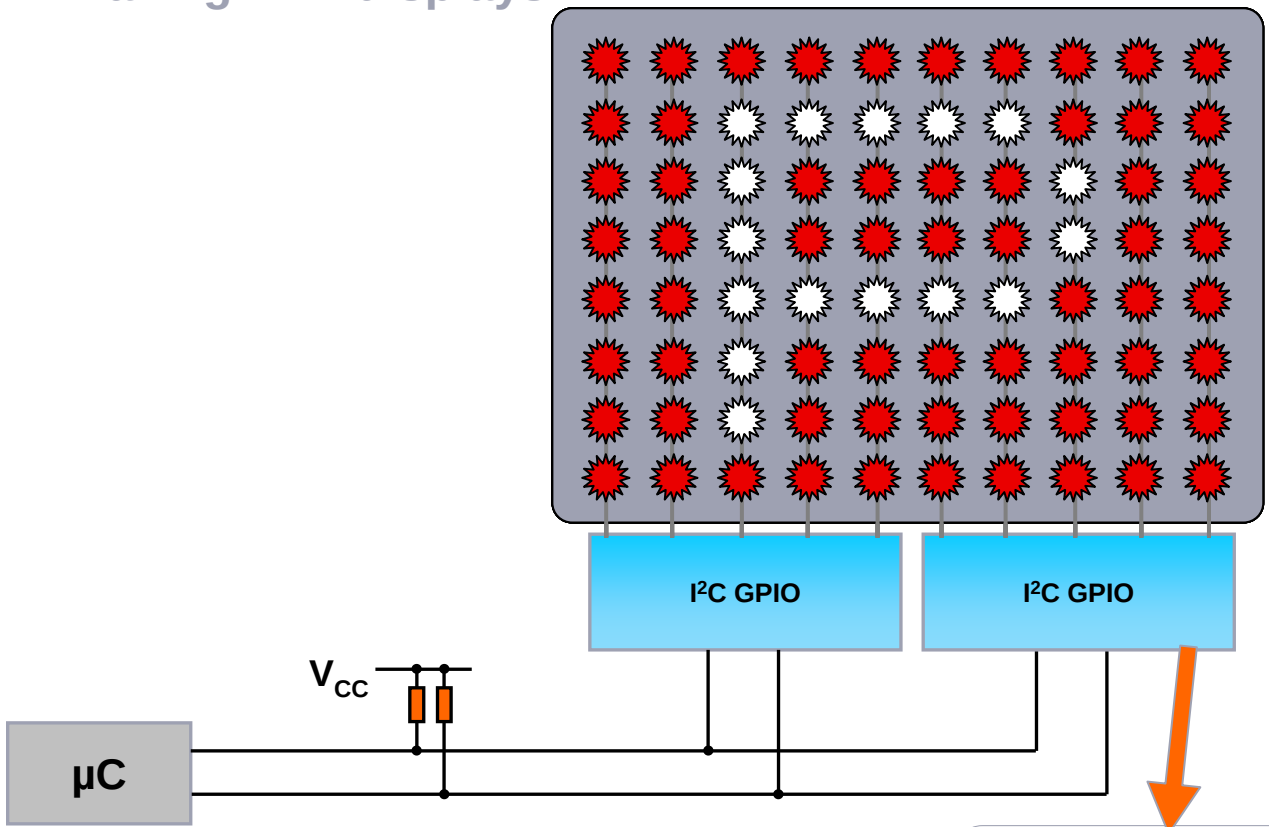
# I²C - GPIOs: General Purpose I/O Expanders

## Application



# I²C - GPIOs: General Purpose I/O Expanders

## Application: making LED displays



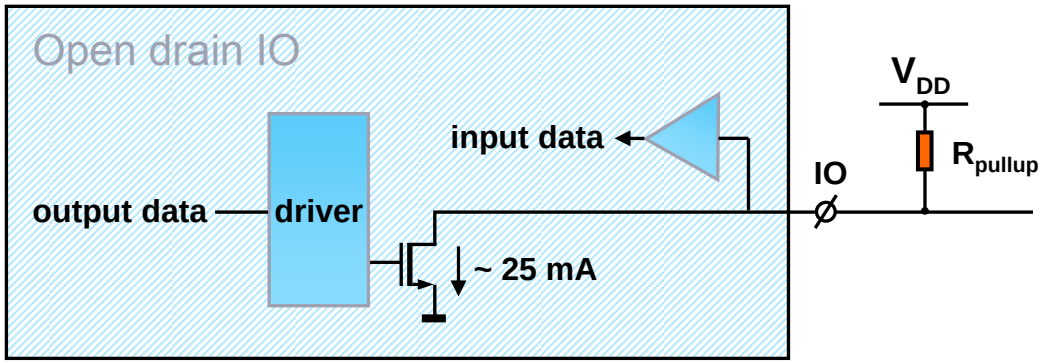

Functions with I²C

Other hardware

Use FM+ devices to increase data rate or to build large LED arrays without I²C buffers

# I²C - GPIOs: General Purpose I/O Expanders

## Intermezzo: Output stages




### Advantages

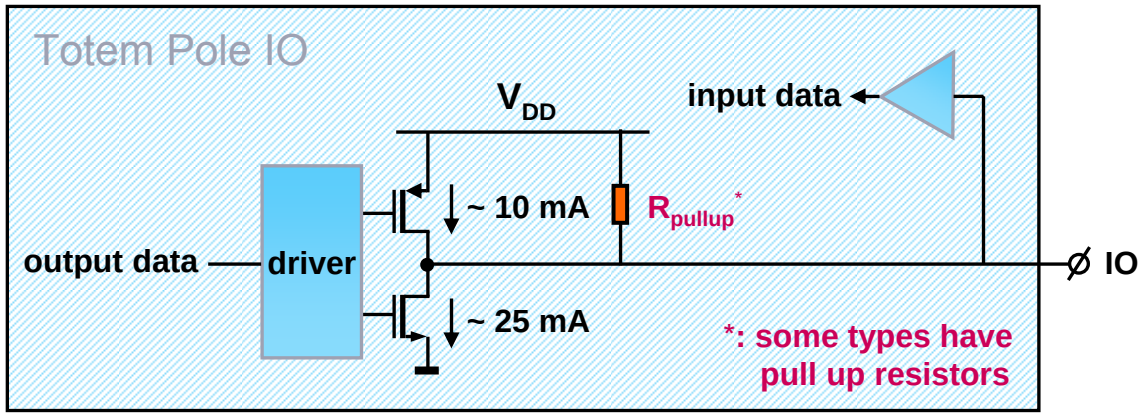
- $V_{DD}$ independent
- 25 mA sink current
- Steep HIGH-LOW transition
- Switch from output to input: only write I/O with “1” to be used as an input

### Disadvantages

- Slow LOW-HIGH transition (depends on the load and on $R_{pullup}$)

# I²C - GPIOs: General Purpose I/O Expanders

## Intermezzo: Output stages (2)




### Advantages

- 25 mA sink current, 10 mA source current
- Steep HIGH-LOW transition
- Steep LOW-HIGH transition

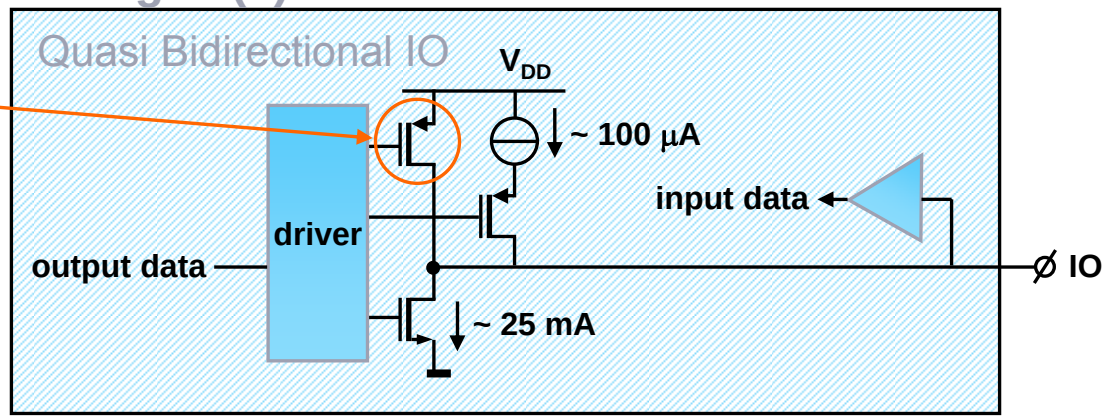
### Disadvantages

- $V_{DD}$ dependent
- Switch from output to input: program the configuration register accordingly (extra programming involved to use the I/O as input)

# I²C - GPIOs: General Purpose I/O Expanders

## Intermezzo: Output stages (3)

Strong PMOS on ONLY 2 osc. clock periods




### Advantages

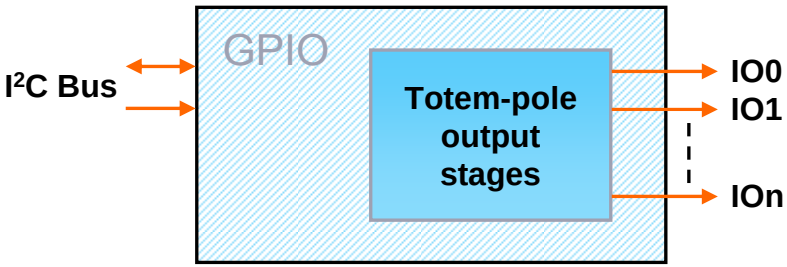
- 25 mA sink current; strong sourcing current only during low-high transition; weak sourcing in steady state
- Steep HIGH-LOW-transition Steep LOW-HIGH transition
- Switch from output to input: only write I/O with “1” to be used as an input
- Usage at higher speeds
- Usage at heavily loaded outputs

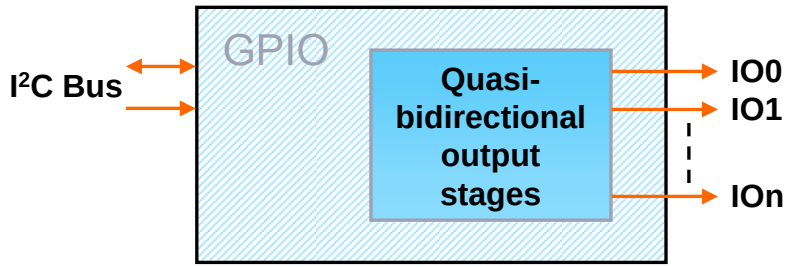
### Disadvantages

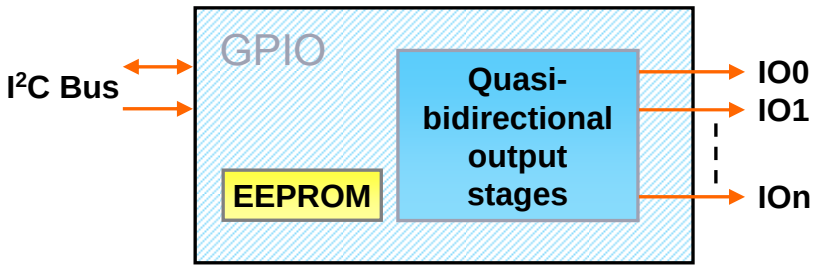
- $V_{DD}$ dependent

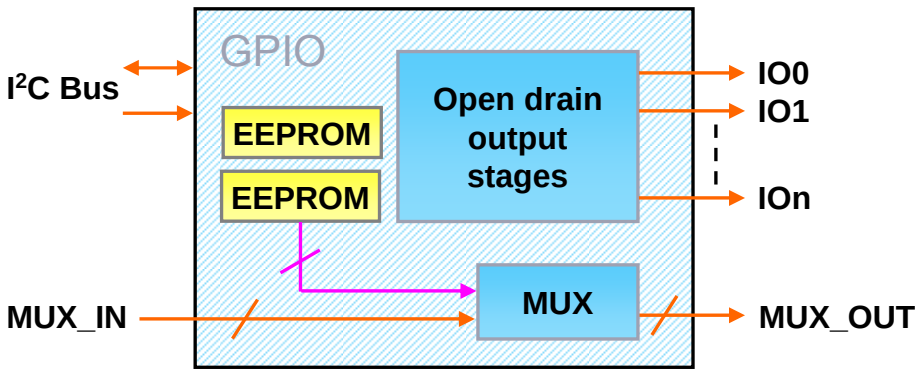
# I²C - GPIOs: General Purpose I/O Expanders

## Product overview




Differentiating GPIO features

- Active-LOW Interrupt output
- Active-LOW Reset input
- Programmable I²C address pins
- Low power consumption
- Additional:
  - EEPROM
  - DIP switch

# I²C - GPIOs: General Purpose I/O Expanders

## PCF8574(A) and PCF8575(C) or PCA type (low cost)

- 8 or 16 I/O pins are used as inputs or outputs
- Open-drain interrupt output
  - activated when any input state differs from its corresponding input port register state
  - Used to indicate to the system master that an input state has changed.
- The PCF8574, PCF8575, and PCF8575C have the same I²C address while the PCF8574A has a slightly different fixed address allowing 16 devices (eight PCF8574A and eight PCF8574/PCF8575(C) in any combination) to be on the same I2C bus.
- The PCF8575 is optimized for automotive applications.
- PCA8574/75 = LOW cost version of the PCF8574/75 (General: not all PCA types are cost down versions)

# I²C - GPIOs: General Purpose I/O Expanders

## PCA9698

- 40-bit I/O advanced expander
- Maximum I²C frequency is 1000 kHz
- SDA output current is stronger than normal to allow use in large area networks without the need for bus buffers
- Each byte will change on acknowledge or wait for a stop condition, allowing you to update the output registers on multiple devices and then change all the outputs at once—which is great for large sign applications
- Each pin can sink 25mA and all pins can be on (40 x 25 mA = 1 A) when using the HVQFN package
- When used for sensing input, the registers can be latched so that you are able to read spurious signals or the inputs can be masked so you don't generate an interrupt
- Three address pins allow for 64 different addresses by using inputs of GND, $V_{DD}$, SCL, or SDA so you can control up to 2560 LEDs on one I²C bus
- There is a GPIO All Call fixed address that allows you to turn on or off LEDs on all the GPIOs in the circuit with a single command sequence—which is great for lamp test or darking the array between displays
- Hardware Reset input will reset the device to its power up state
- Output Enable pin that will tri-state the outputs which are selectable between push pull and open drain

# I²C - GPIOs: General Purpose I/O Expanders

## Product summary

| Type nr. | # of Outputs | Quasi Output | Push-pull Output | Internal Pull-Up Resistor/ current source | Vcc Range (V) | 5 V Tolerance | I²C Max Frequency [kHz] | SPI Max Frequency [MHz] | ‾OE | Hardware reset | Interrupt Output | 2 kbit EEPROM | 5-bit Multiplex / 1 Bit latch EEPROM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PCA9500 | 8 | ✓ | | ✓ | 2.5-3.6 | ✓ | 400 | | | | | ✓ | |
| PCA9501 | 8 | ✓ | | ✓ | 2.5-3.6 | ✓ | 400 | | | | ✓ | ✓ | |
| PCA9558 | 8 | Open Drain | | | 2.3-5.5 | | 400 | | | | | ✓ | ✓ |
| PCA9505/06 | 40 | Open Drain | ✓ | | 2.3-5.5 | ✓ | 400 | | ✓ | ✓ | ✓ | | |
| PCF8574(A) | 8 | ✓ | | ✓ | 2.5-6.0 | ✓ | 100 | | | | ✓ | | |
| PCA8574(A)* | 8 | ✓ | | ✓ | 2.3-5.5 | ✓ | 400 | | | | ✓ | | |
| PCF8575 | 16 | ✓ | | ✓ | 2.5-5.5 | ✓ | 400 | | | | ✓ | | |
| PCF8575C | 16 | Open Drain | | | 4.5-5.5 | ✓ | 400 | | | | ✓ | | |
| PCA9670* | 8 | ✓ | | ✓ | 2.3-5.5 | ✓ | 1000 | | | ✓ | | | |
| PCA9672* | 8 | ✓ | | ✓ | 2.3-5.5 | ✓ | 1000 | | | ✓ | ✓ | | |
| PCA9674(A)* | 8 | ✓ | | ✓ | 2.3-5.5 | ✓ | 1000 | | | | ✓ | | |
| PCA9671* | 16 | ✓ | | ✓ | 2.3-5.5 | ✓ | 1000 | | | ✓ | | | |
| PCA9673* | 16 | ✓ | | ✓ | 2.3-5.5 | ✓ | 1000 | | | ✓ | ✓ | | |
| PCA9675* | 16 | ✓ | | ✓ | 2.3-5.5 | ✓ | 1000 | | | | ✓ | | |
| PCA8575* | 16 | ✓ | | ✓ | 2.3-5.5 | ✓ | 400 | | | | ✓ | | |

*: GPIOs Recommended for LED Applications

# I²C - GPIOs: General Purpose I/O Expanders

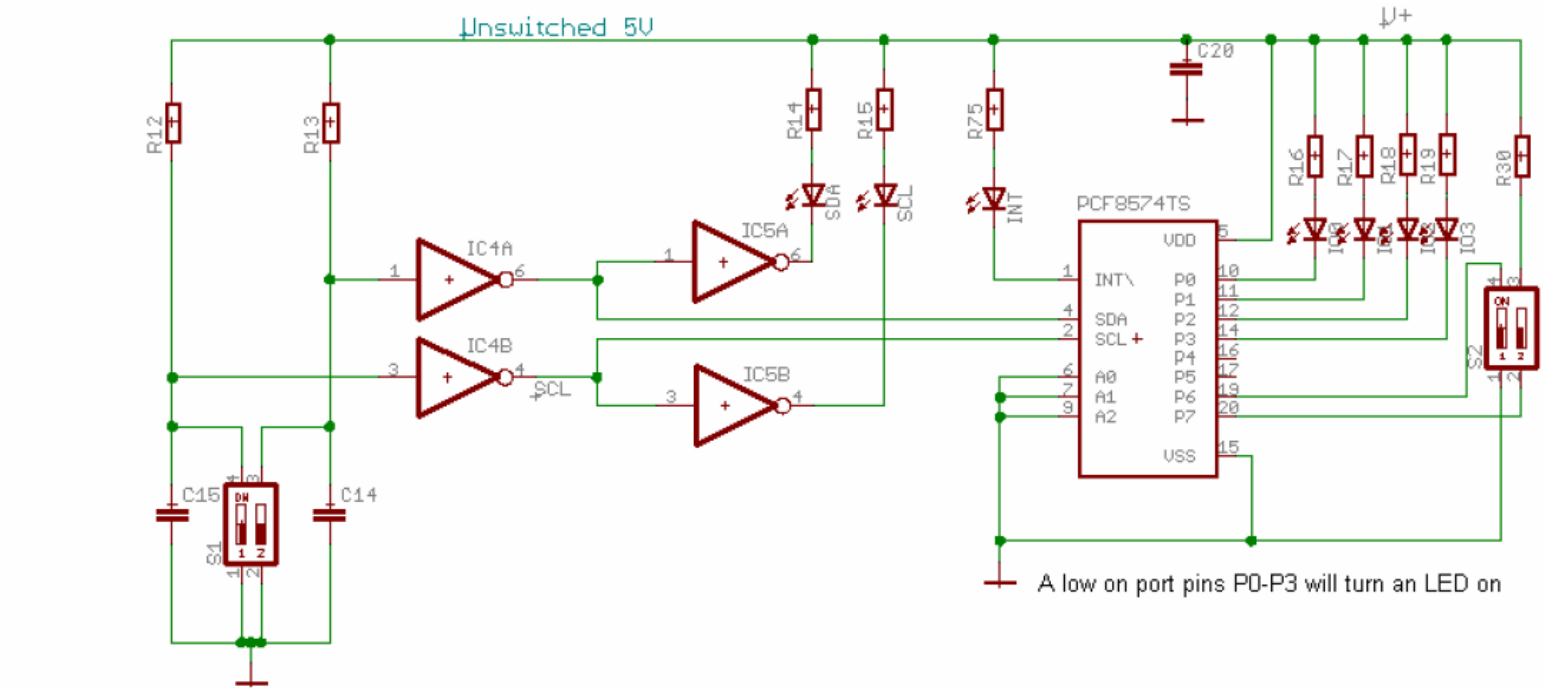
## Product summary (2)

| Type nr. | # of Outputs | Quasi Output | Push-pull Output | Internal Pull-Up Resistor/ current source | Vcc Range (V) | 5 V Tolerance | I²C Max Frequency [kHz] | SPI Max Frequency [MHz] | $\overline{OE}$ | Hardware reset | Interrupt Output |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PCA9502 | 8 | | ✓ | | 2.3-3.6 | ✓ | 400 | 15 | | ✓ | ✓ |
| PCA9505* | 40 | | ✓ | ✓ | 2.3-5.5 | ✓ | 400 | | ✓ | ✓ | ✓ |
| PCA9506* | 40 | | ✓ | | 2.3-5.5 | ✓ | 400 | | ✓ | ✓ | ✓ |
| PCA9534 | 8 | | ✓ | | 2.3-5.5 | ✓ | 400 | | | | ✓ |
| PCA9535 | 16 | | ✓ | | 2.3-5.5 | ✓ | 400 | | | | ✓ |
| PCA9536* | 4 | | ✓ | ✓ | 2.3-5.5 | ✓ | 400 | | | | |
| PCA9537* | 4 | | ✓ | | 2.3-5.5 | ✓ | 400 | | | ✓ | ✓ |
| PCA9538* | 8 | | ✓ | | 2.3-5.5 | ✓ | 400 | | | ✓ | ✓ |
| PCA9539* | 16 | | ✓ | | 2.3-5.5 | ✓ | 400 | | | ✓ | ✓ |
| PCA9554(A) | 8 | | ✓ | ✓ | 2.3-5.5 | ✓ | 400 | | | | ✓ |
| PCA9555 | 16 | | ✓ | ✓ | 2.3-5.5 | ✓ | 400 | | | | ✓ |
| PCA9557* | 8 | | ✓ | | 2.3-5.5 | ✓ | 400 | | | ✓ | |
| PCA9698* | 40 | Open Drain | ✓ | | 2.3-5.5 | ✓ | 1000 | | ✓ | ✓ | ✓ |

*: GPIOs Recommended for LED Applications

# I²C - GPIOs: General Purpose I/O Expanders

## Demoboard I2C 2005-1: PCF8574 schematic



# I²C - GPIOs: General Purpose I/O Expanders

## Demoboard I2C 2005-1: PCF8574 GUI

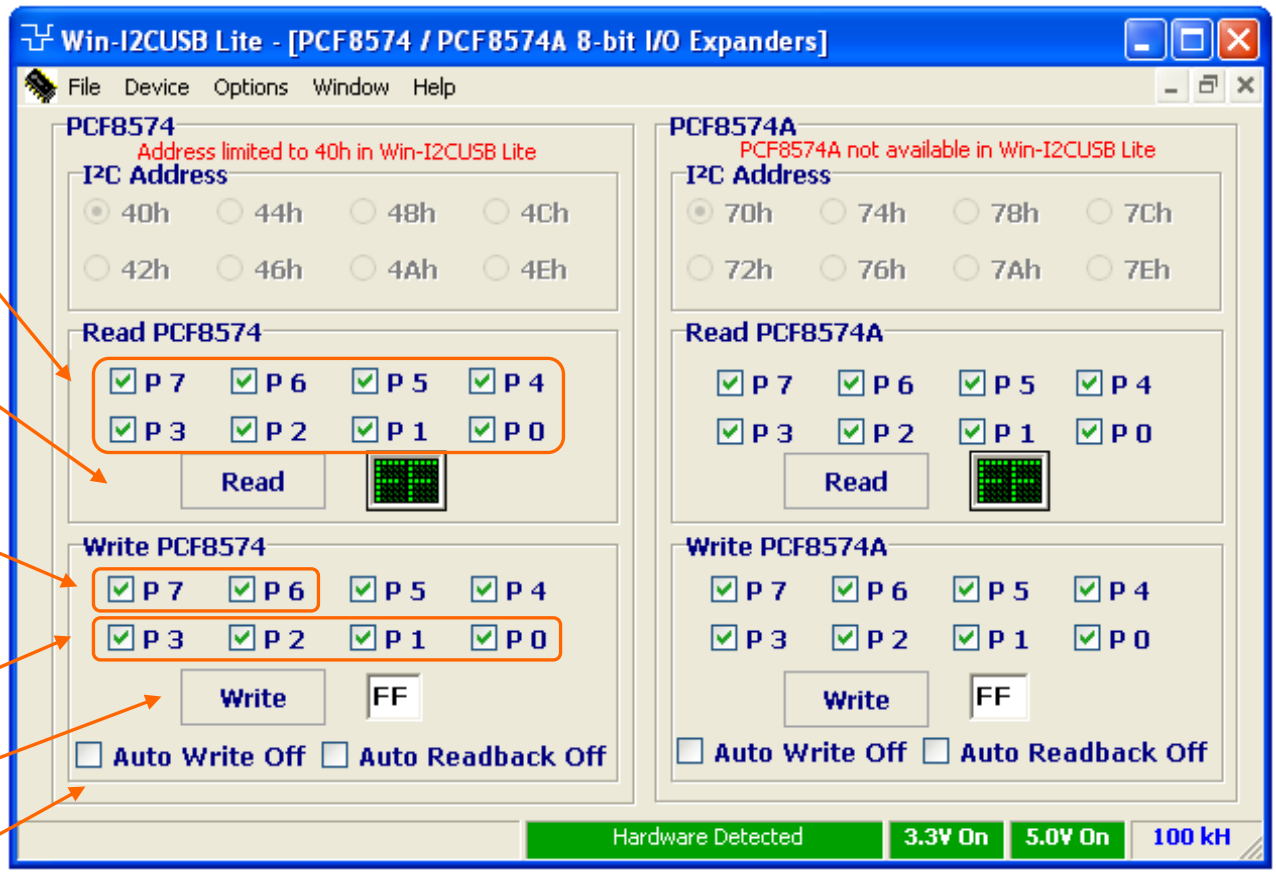
Checkboxes (read-only) represent state of port pins
Checked = High
Unchecked = Low

Press the Read button to find the state of the IO pins

P6 = switch connected to ground
P7 = switch to Vcc

Pins P0-P3 control LEDs
Checked = high (LED off)
Unchecked = low (LED on)

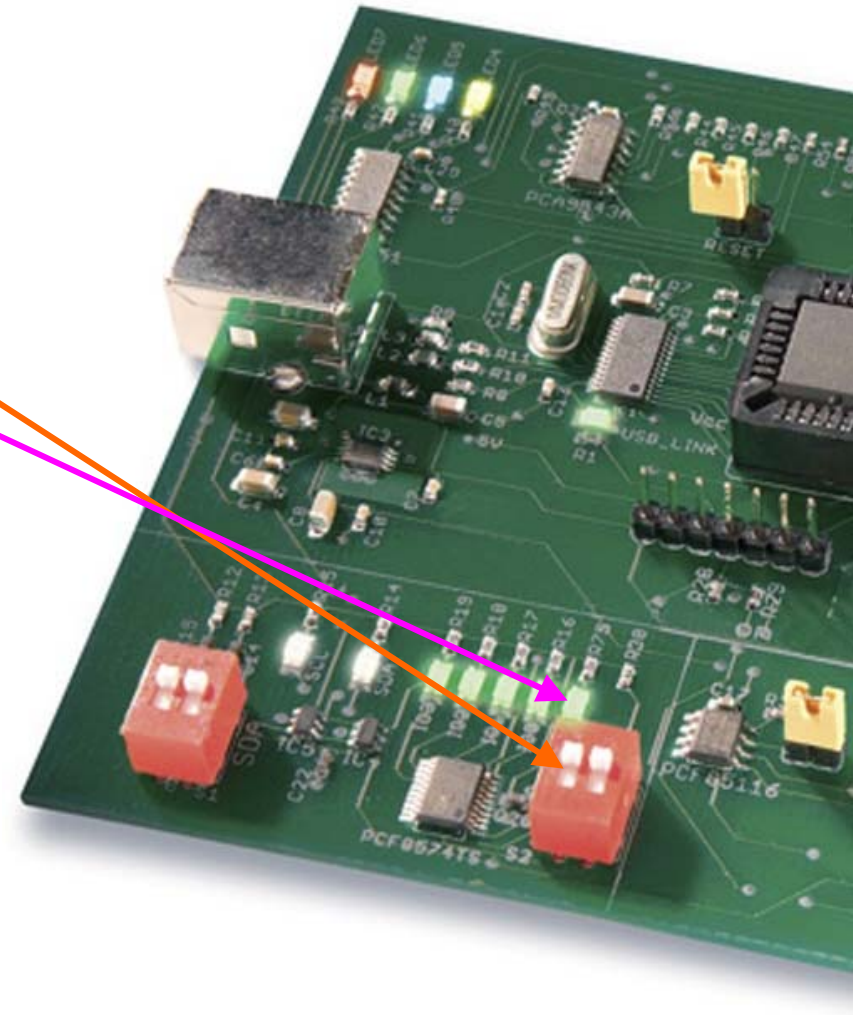
Press Write button to send data to PCF8574 or use Auto Write to send automatically



# I²C - GPIOs: General Purpose I/O Expanders

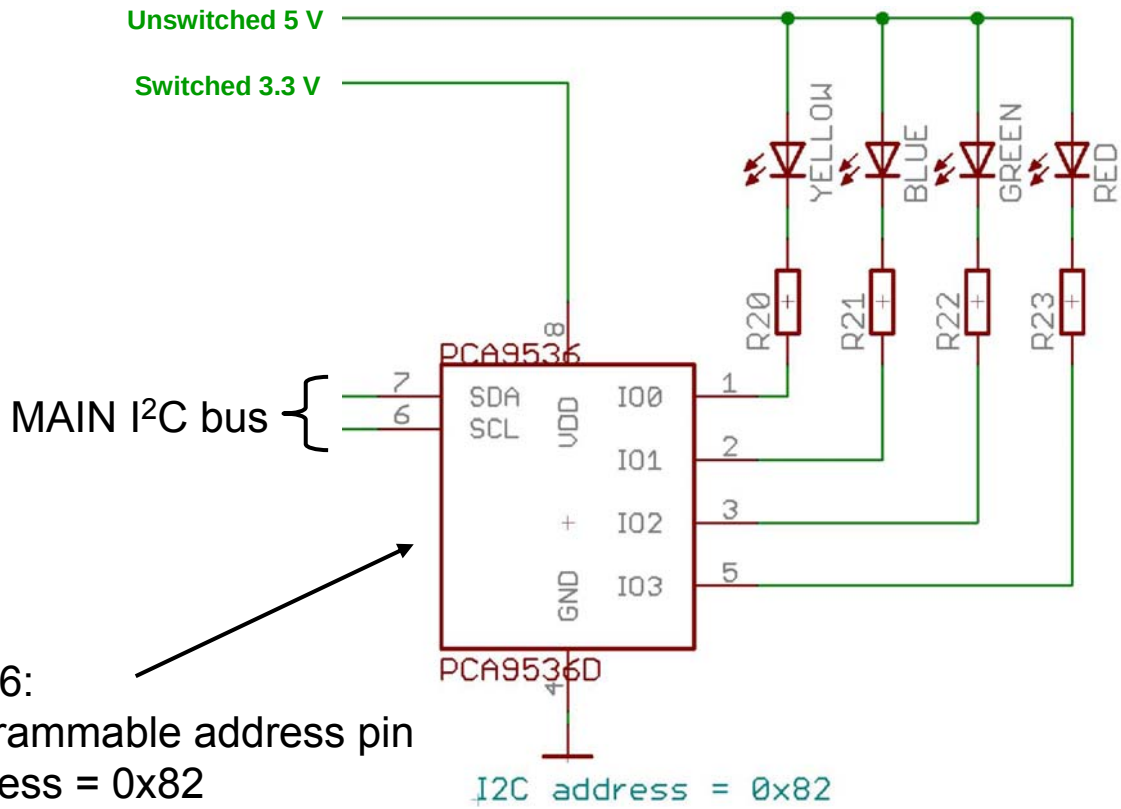
## Demoboard I2C 2005-1: Exercise: Understanding quasi-bidirectional outputs

1. Set all the port pins high using Win-I²CUSB Lite.
2. Change the state of switch 1 on S2.
3. The INT LED should illuminate.
4. Change the state of the switch. The LED should turn off.
5. Turn the INT LED again by changing the switch.
6. Using Win-I2CUSB Lite. Read the PCF8574. What happened to the Interrupt?
7. Turn the INT LED again by changing the switch. Write to the PCF8574. What happened to the Interrupt?



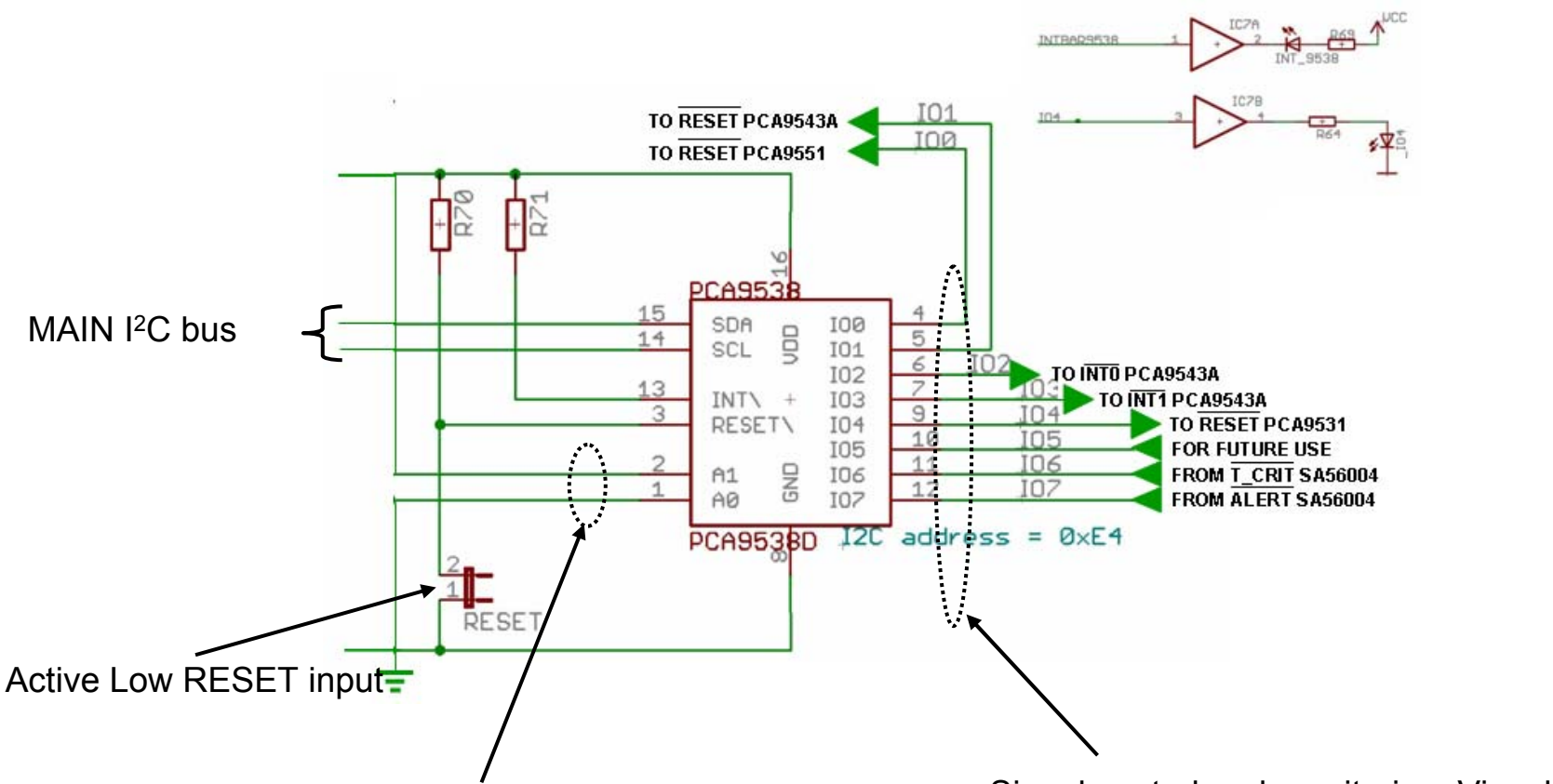
# I²C - GPIOs: General Purpose I/O Expanders

## Demoboard I2C 2005-1: PCA9536 schematic



# I²C - GPIOs: General Purpose I/O Expanders

## Demoboard I2C 2005-1: PCA9538 schematic



# I²C - GPIOs: General Purpose I/O Expanders

## Demoboard I2C 2005-1: PCA9538 GUI

I²C address:
PCA9536 = 0x82
PCA9538 = 0xE4

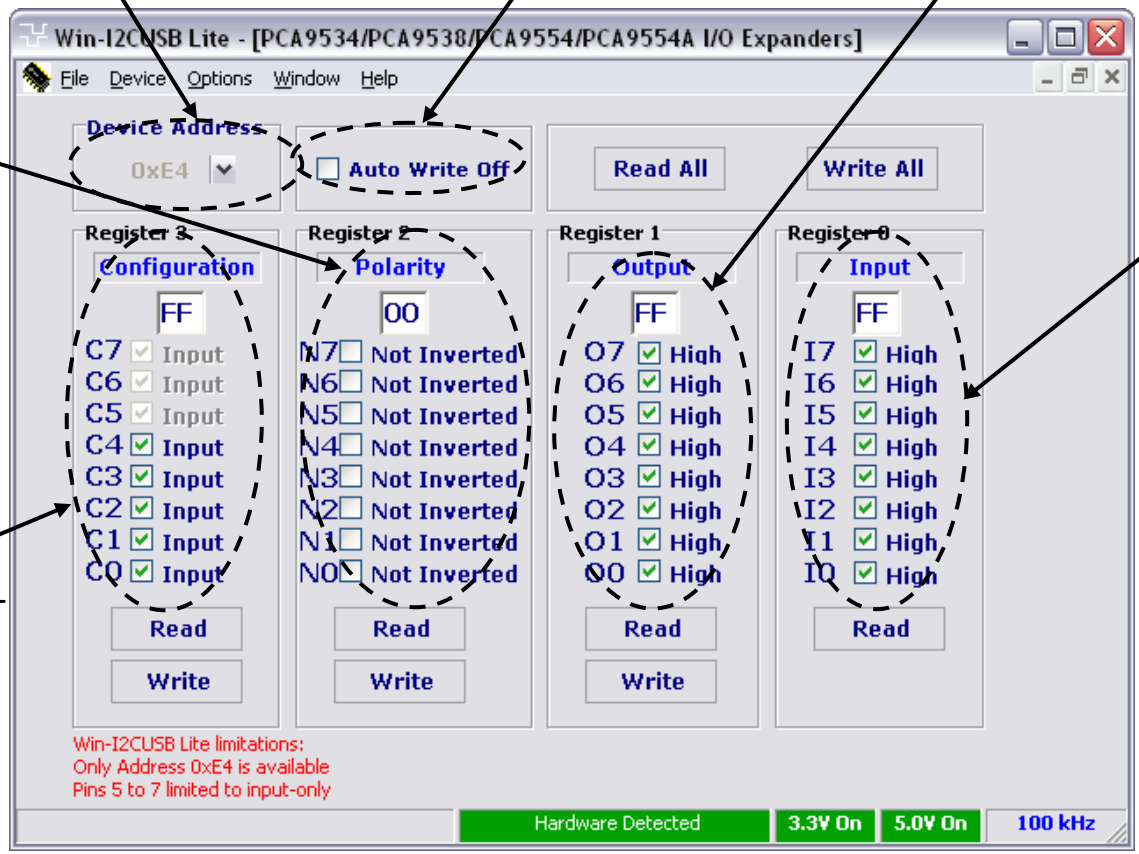
Auto Write Feature

Output register programming the logic state of the pins programmed as inputs

Polarity register- allowing to invert or not the logic values read from the pins

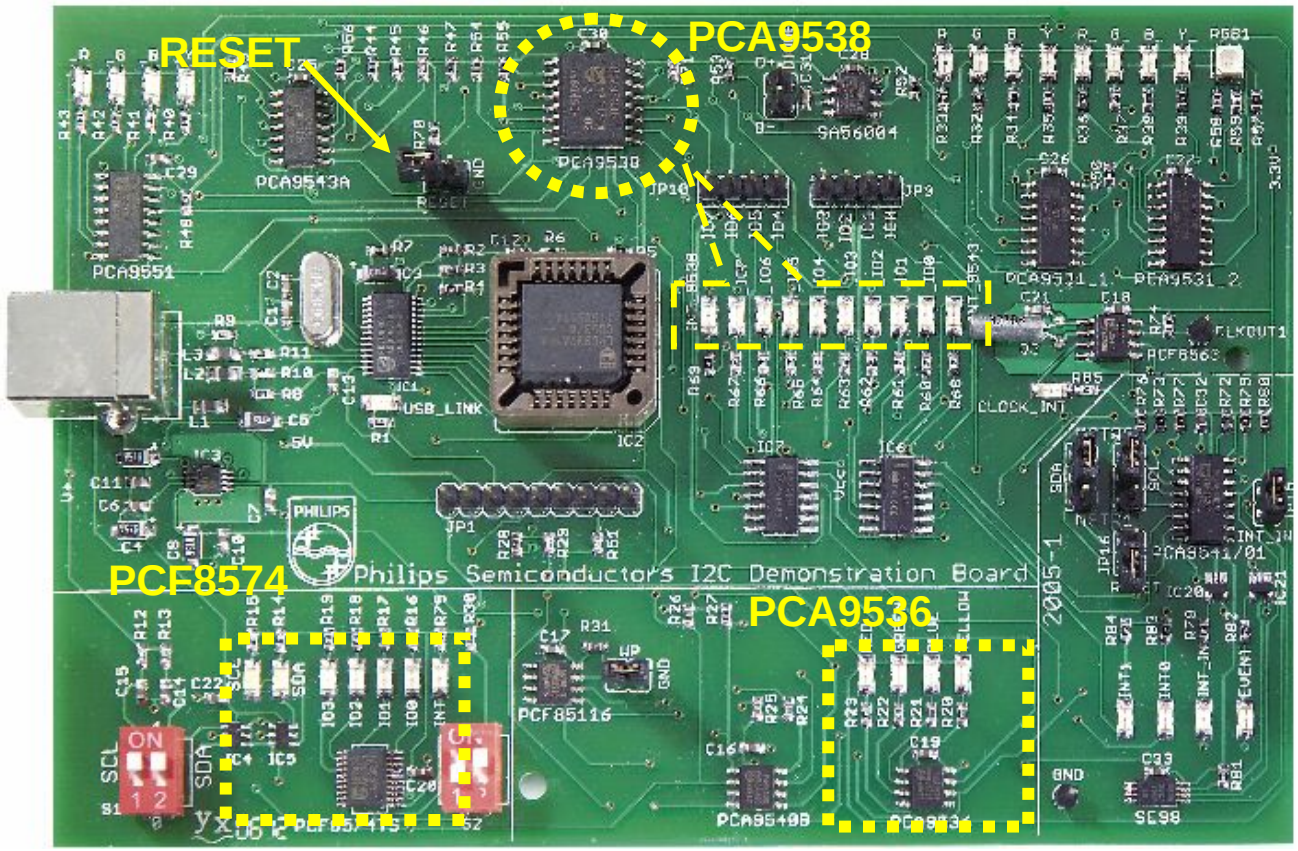
Input register reading the IOs logic states

Configuration register- programming IOs as inputs or output



# I²C - GPIOs: General Purpose I/O Expanders

## Demoboard I2C 2005-1: PCA9536, PCA9538 on the board



# I²C - GPIOs: General Purpose I/O Expanders

## Demoboard I2C 2005-1: Exercise A: PCA9536

1. Power sequence the board → Board OFF then ON
2. Open the PCA9536 GUI:
   Device → IO Expanders → PCA9536 4-bit I/O Expander
3. Switch ON the 4 LEDs connected to the PCA9536
   Configuration register: the 4 I/Os need to be configured as outputs (checked)
   Output register: the 4 outputs need to be forced Low
4. Uncheck C0, C1, C2 and C3
   Check O0 and O1 in the Output Register (O2 and O3 unchecked)
   Check N0 and N2 in the Polarity Register (N1 and N3 unchecked)
   Read the Input Register and explain what you see

   I0 = Low → Inverted Logic state of I/O0
   I1 = High → Logic state of I/O1
   I2 = High → Inverted Logic state of I/O2
   I3 = Low → Logic state of I/O3

# I²C - GPIOs: General Purpose I/O Expanders

## Demoboard I2C 2005-1: Exercise B: PCA9538

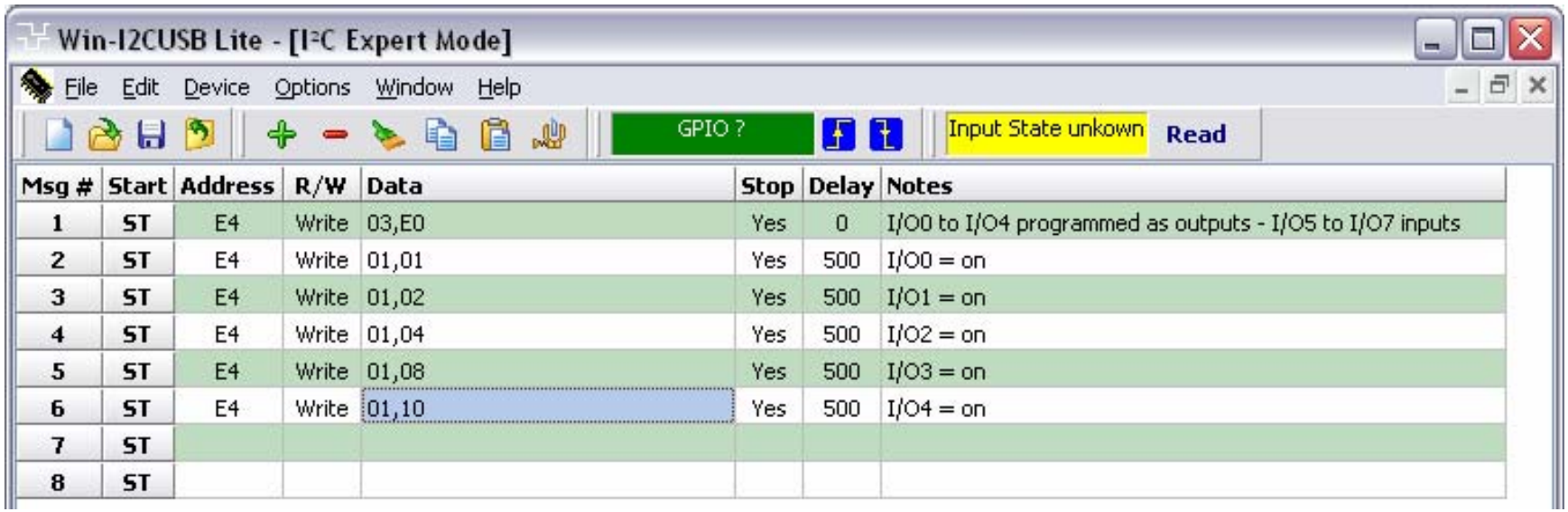
Write the following I²C sequence (Red) using the Expert Mode

1. Power sequence the board → Board OFF then ON
2. Program PCA9538 I/O0 to I/O4 as output (and be sure that I/O5 to I/O7 are configured as inputs)
3. Switch ON PCA9538 I/O0 only and wait 0.5 s
4. Switch ON PCA9538 I/O1 only and wait 0.5 s
5. Switch ON PCA9538 I/O2 only and wait 0.5 s
6. Switch ON PCA9538 I/O3 only and wait 0.5 s
7. Switch ON PCA9538 I/O4 only and wait 0.5 s

Execute it in loop

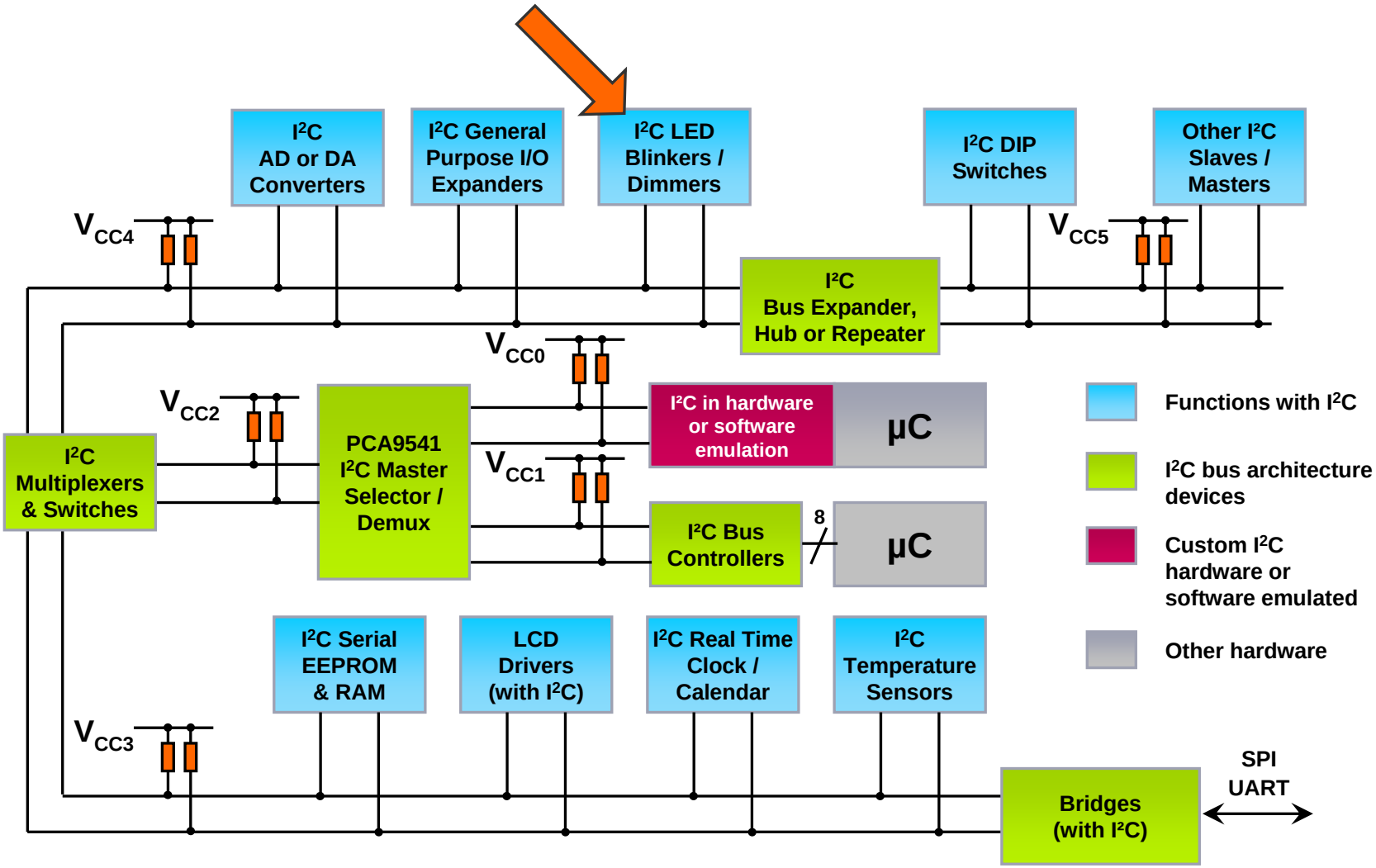
# I²C - GPIOs: General Purpose I/O Expanders

## Demoboard I2C 2005-1: Exercise B: PCA9538; solution



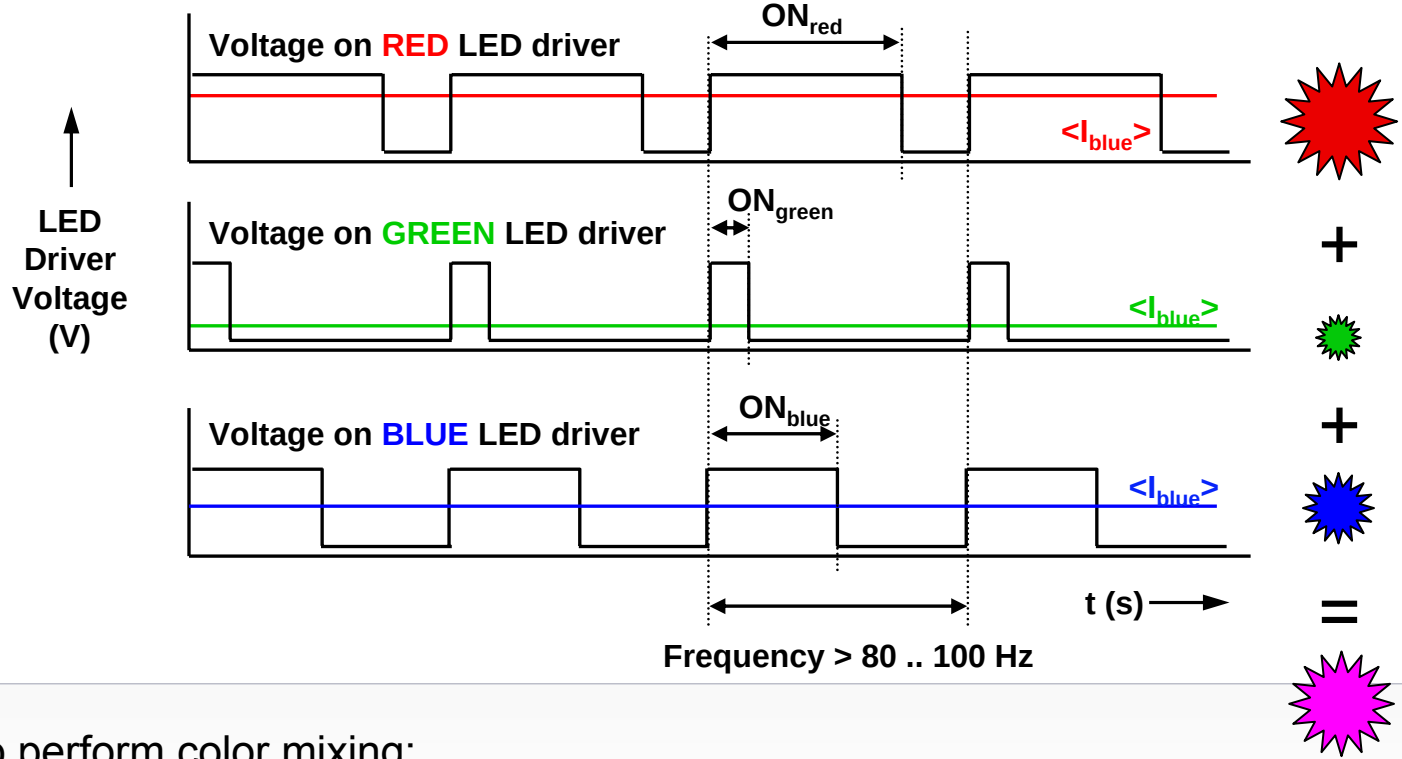

| Msg # | Start | Address | R/W | Data | Stop | Delay | Notes |
|---|---|---|---|---|---|---|---|
| 1 | ST | E4 | Write | 03,E0 | Yes | 0 | I/O0 to I/O4 programmed as outputs - I/O5 to I/O7 inputs |
| 2 | ST | E4 | Write | 01,01 | Yes | 500 | I/O0 = on |
| 3 | ST | E4 | Write | 01,02 | Yes | 500 | I/O1 = on |
| 4 | ST | E4 | Write | 01,04 | Yes | 500 | I/O2 = on |
| 5 | ST | E4 | Write | 01,08 | Yes | 500 | I/O3 = on |
| 6 | ST | E4 | Write | 01,10 | Yes | 500 | I/O4 = on |
| 7 | ST | | | | | | |
| 8 | ST | | | | | | |

# I²C - LED Drivers, Blinkers and Dimmers



# I²C - LED Drivers, Blinkers and Dimmers

## Intermezzo: Using PWM for color mixing in RGB LED




To perform color mixing:

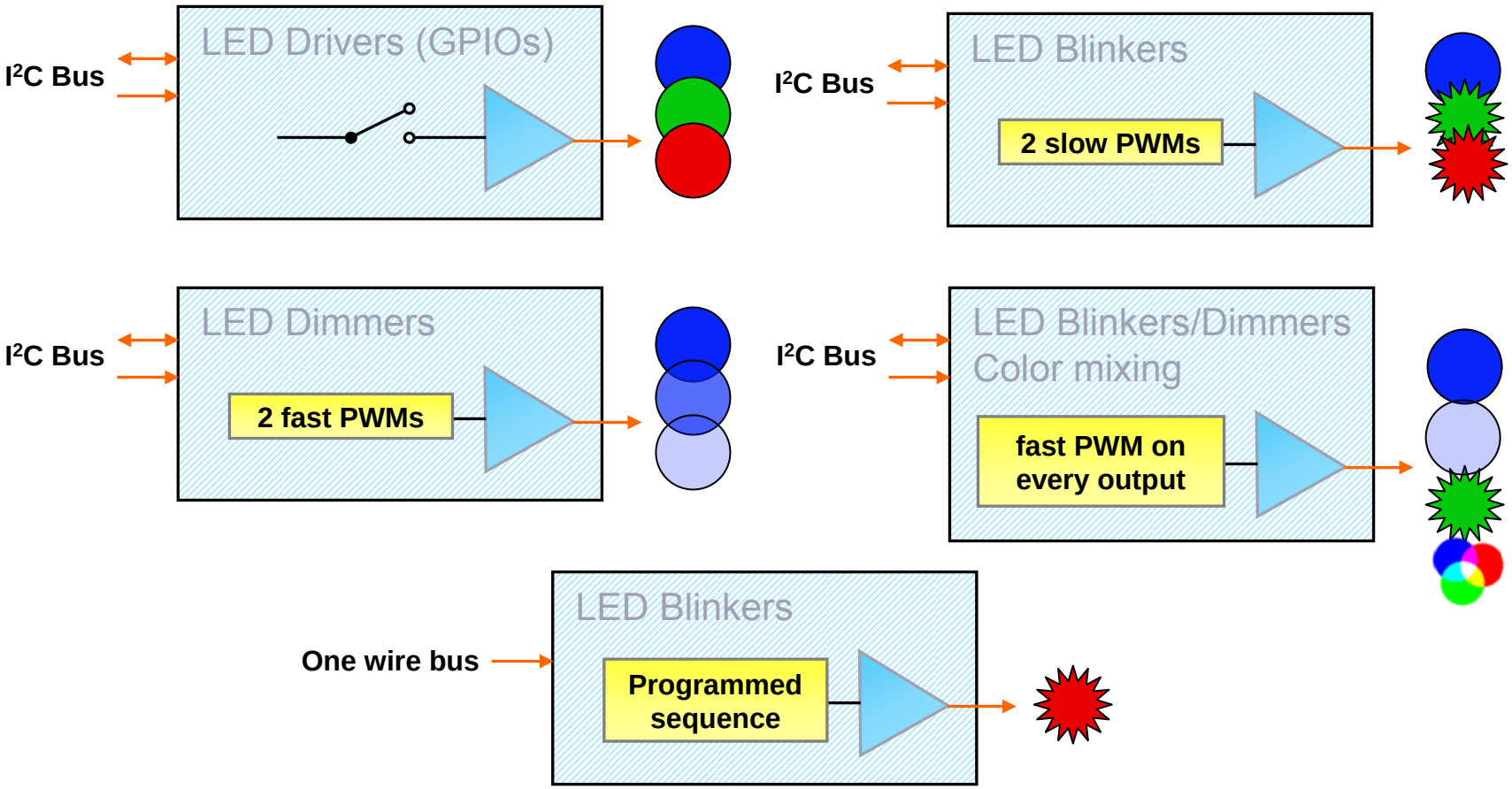
- Frequency high enough so the human eye does not see the ON/OFF phases
- Brightness for each primary color (desired amount of each primary color) is controlled with the duty cycle
- Human eye sees the sum of primary colors' average brightness: X% Red + Y% Green + Z% Blue

# I²C - LED Blinkers, Dimmers and Drivers

## Controllability via I²C; color mixing, brightness, blinking

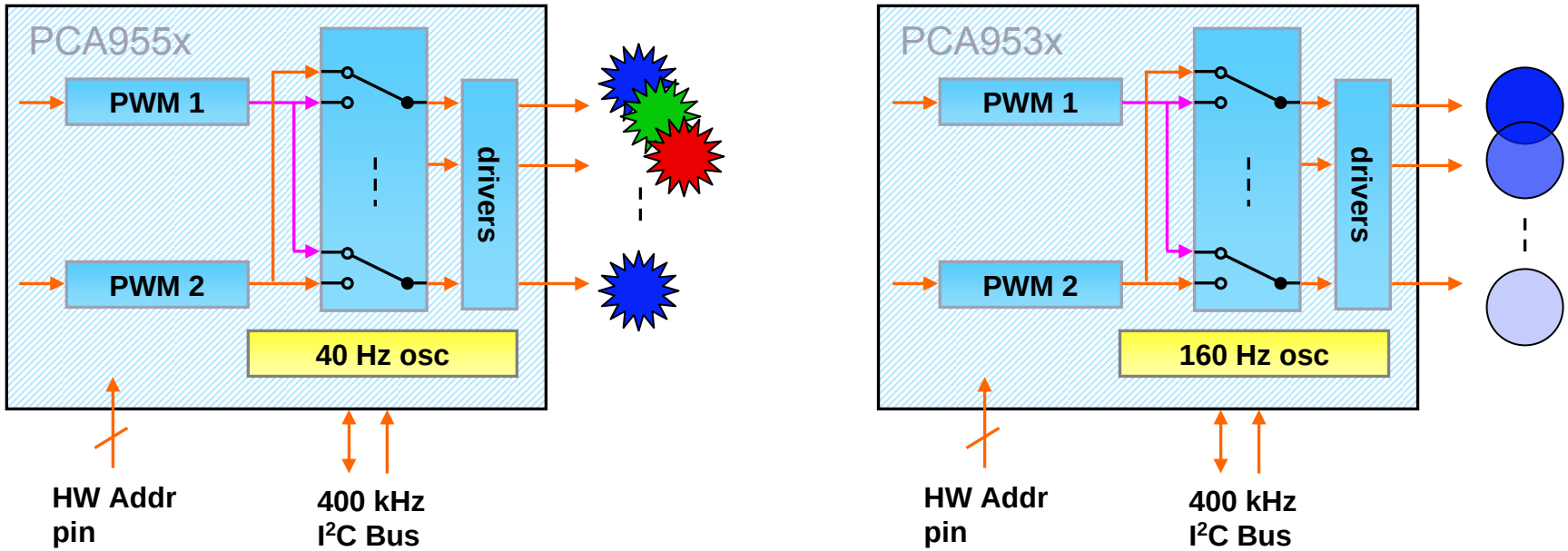
# I²C - LED Drivers, Blinkers and Dimmers

## Product overview



# I²C - LED Drivers, Blinkers and Dimmers

## PCA955x - Blinkers / PCA953x - Dimmers

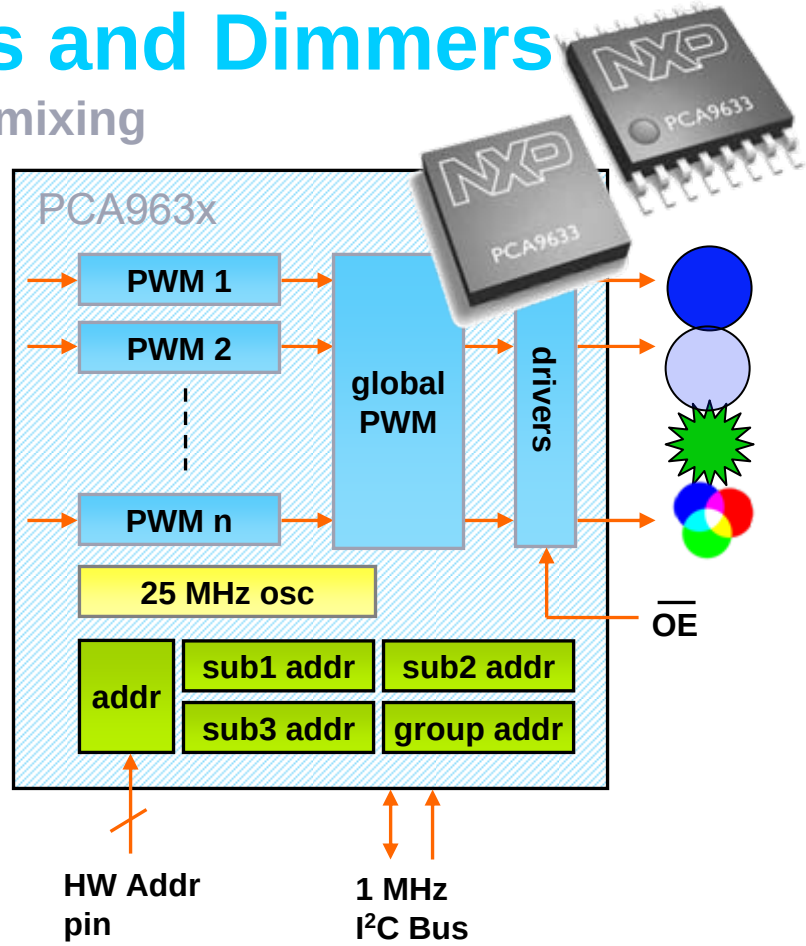



- Drive up to 25 mA per output
- Fully integrated oscillator generates programmed blinking / dimming schemes
- Each LED can programmed to be:
  - Fully OFF, ON
  - Blinking / dimming at frequency1 or at frequency2

# I²C - LED Drivers, Blinkers and Dimmers

## PCA963x - RGBA Dimmer / Blinker / Color mixing

- 4/8/16 channel-LED drivers
  - On / Off
  - Individual PWM control
  - Group PWM Control
- Output Structure
  - 5 V tolerant Outputs
  - Programmable totem-pole / open drain
  - 10 mA source, 25 mA sink capability
- /OE input pin allows hardware dimming / blinking
- Enables groups of devices to be addressed at the same time
  - 1 Group-Call Address
  - 3 Sub-Call Addresses
- Fast Mode Plus (Fm+) compatible I²C protocol
- $V_{DD}$ = 2.3 V to 5.5 V



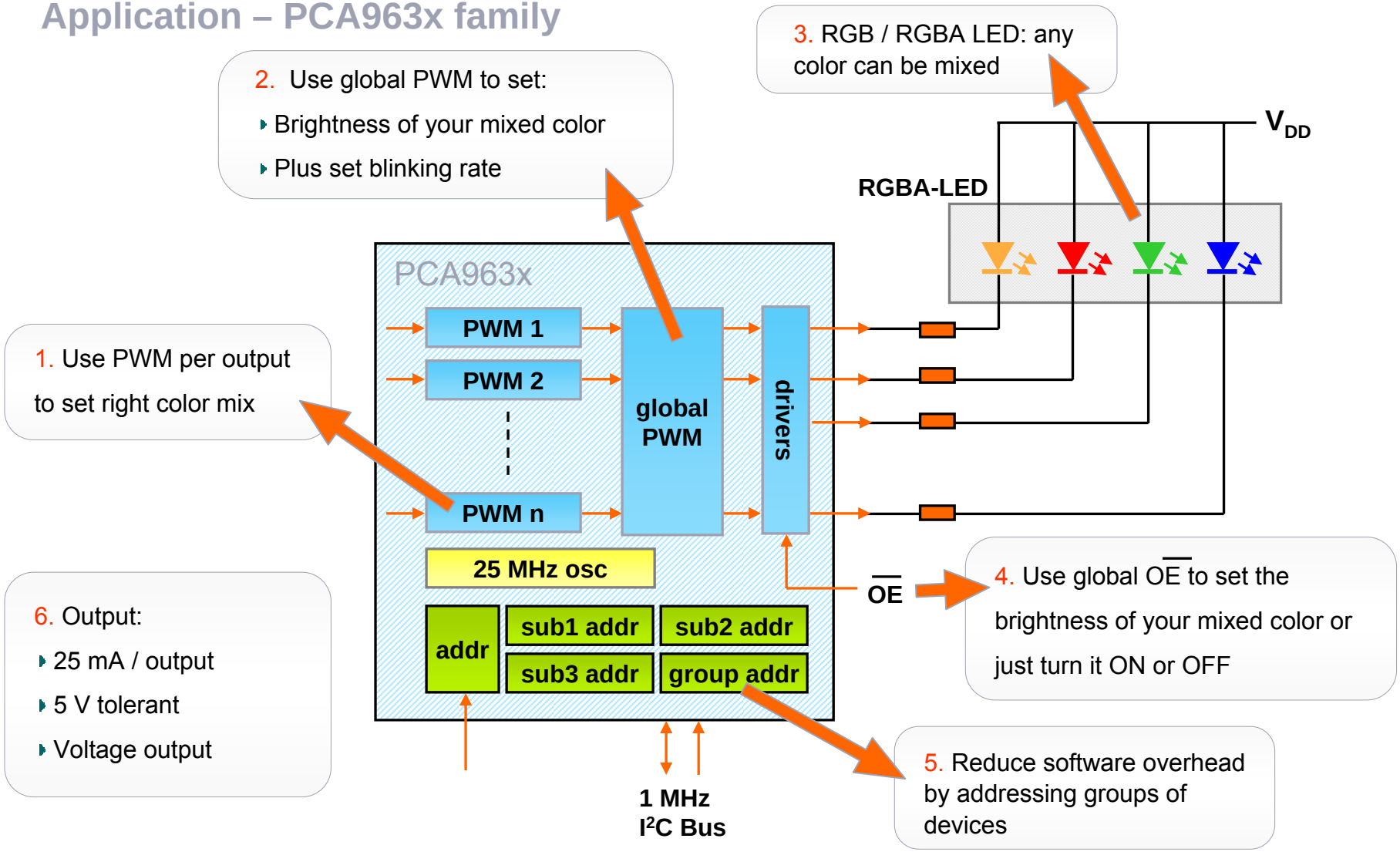

**PCA9633 8-pin = no HW addr pin**

**PCA9633 10-pin = 2 HW addr pin**

**PCA9633 16-pin = 7 HW addr pin**

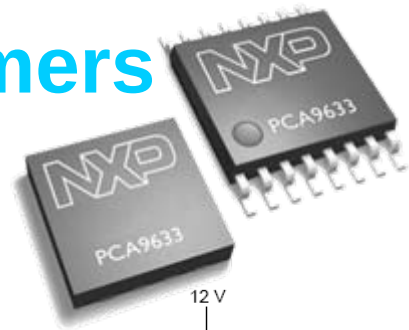
# I²C - LED Blinkers, Dimmers and Drivers

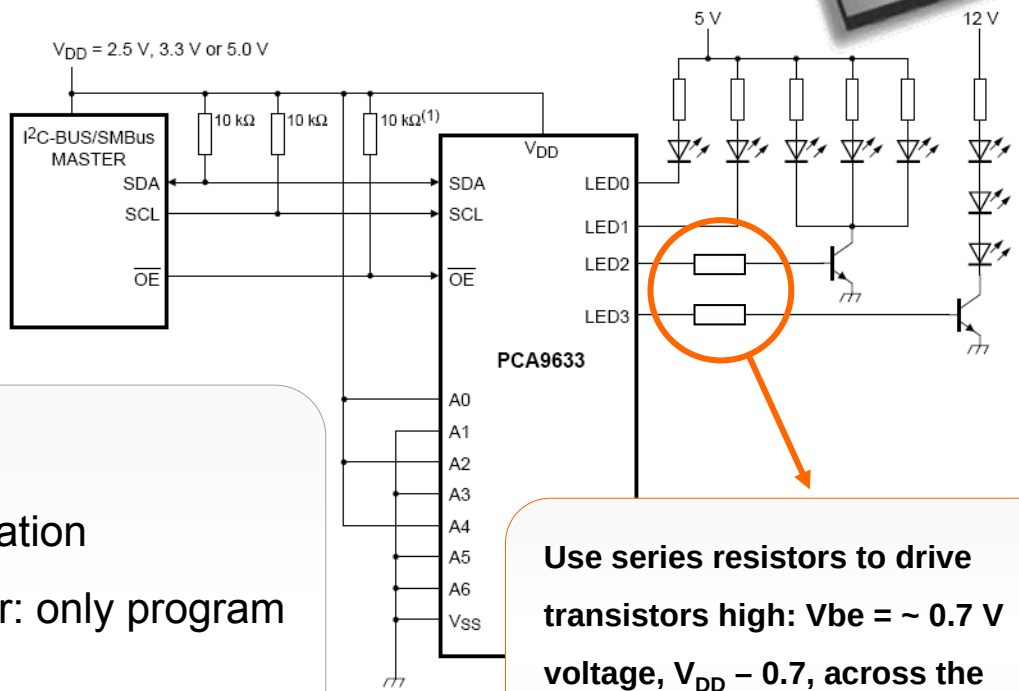
## Application – PCA963x family



# I²C - LED Drivers, Blinkers and Dimmers

## Application



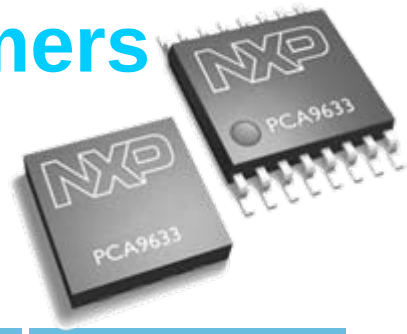




- Drive RGBA LEDs
- Distribution of power dissipation
- Off-load the microprocessor: only program LED-controller once
- Outputs of blinkers / dimmers not used for LED driving can be used as GPI/O (not all devices)

**Use series resistors to drive transistors high: Vbe = ~ 0.7 V voltage, $V_{DD}$ – 0.7, across the series resistor**

# I²C - LED Drivers, Blinkers and Dimmers

## Product summary

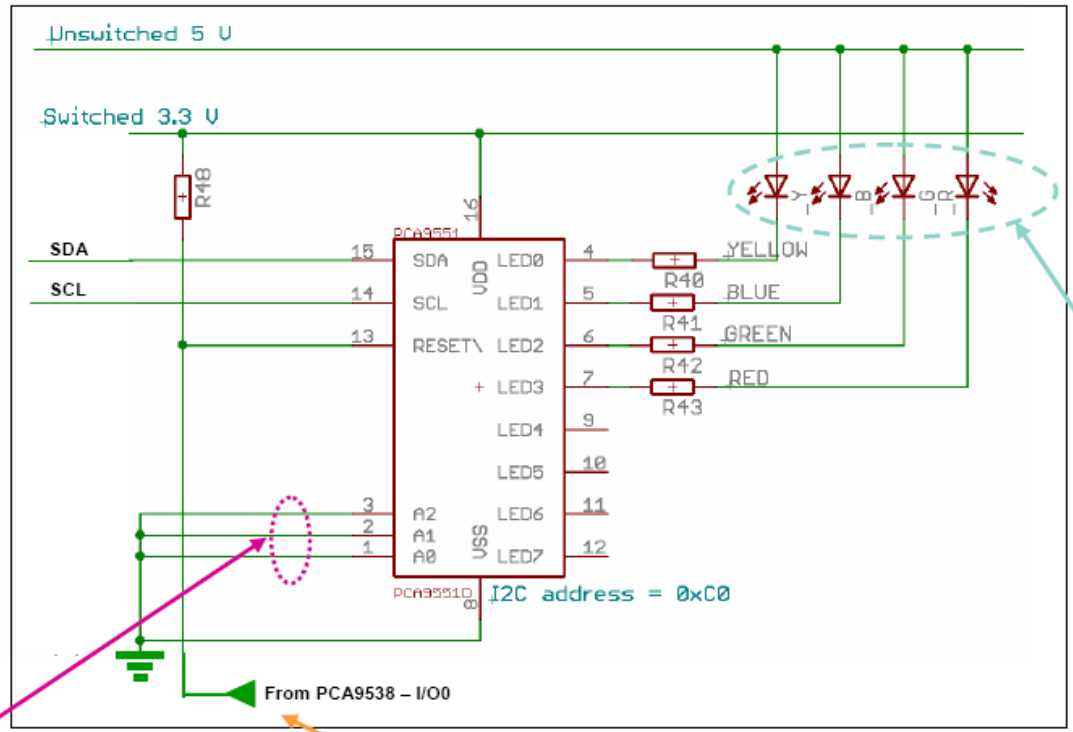



| | Frequency | Duty Cycle | Resolution | Type nr. |
|---|---|---|---|---|
| Drivers | On/off | - | | See GPIOs |
| Blinkers | 40 Hz - 1/6.4 s | 100 - 0.4% | 8-bit | PCA955x |
| Dimmers | 160 Hz - 1/1.6 s | 0 - 99.6% | 8-bit | PCA953x |
| Blinkers/dimmers | 24 Hz - 1/10.73 s | 0 - 99.6% | 8-bit | PCA963x |

| # of outputs | LED Blinkers | LED Dimmers | LED Dimmers / Blinkers |
|---|---|---|---|
| 2 | PCA9550 | PCA9530 | |
| 4 | PCA9553 | PCA9533 | PCA9633 **PCA9632 (low power)** |
| 8 | PCA9551 | PCA9531 | PCA9634 |
| 16 | PCA9552 | PCA9532 | PCA9635 |

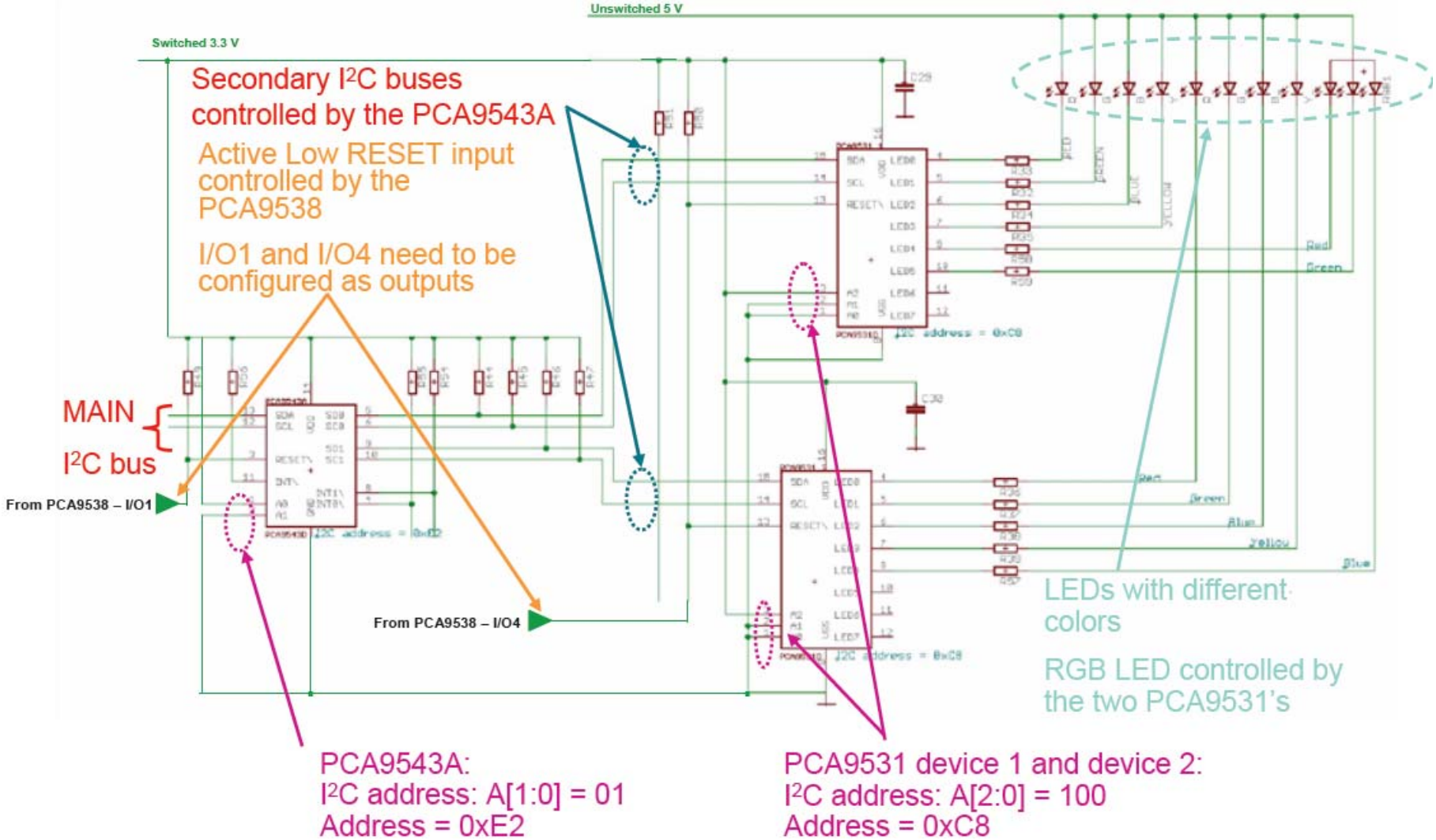
# I²C - LED Drivers, Blinkers and Dimmers

## Demoboard I2C 2005-1: PCA9551 (blinker) schematic



# I²C - LED Drivers, Blinkers and Dimmers

## Demoboard I2C 2005-1: PCA9531 (dimmer) schematic



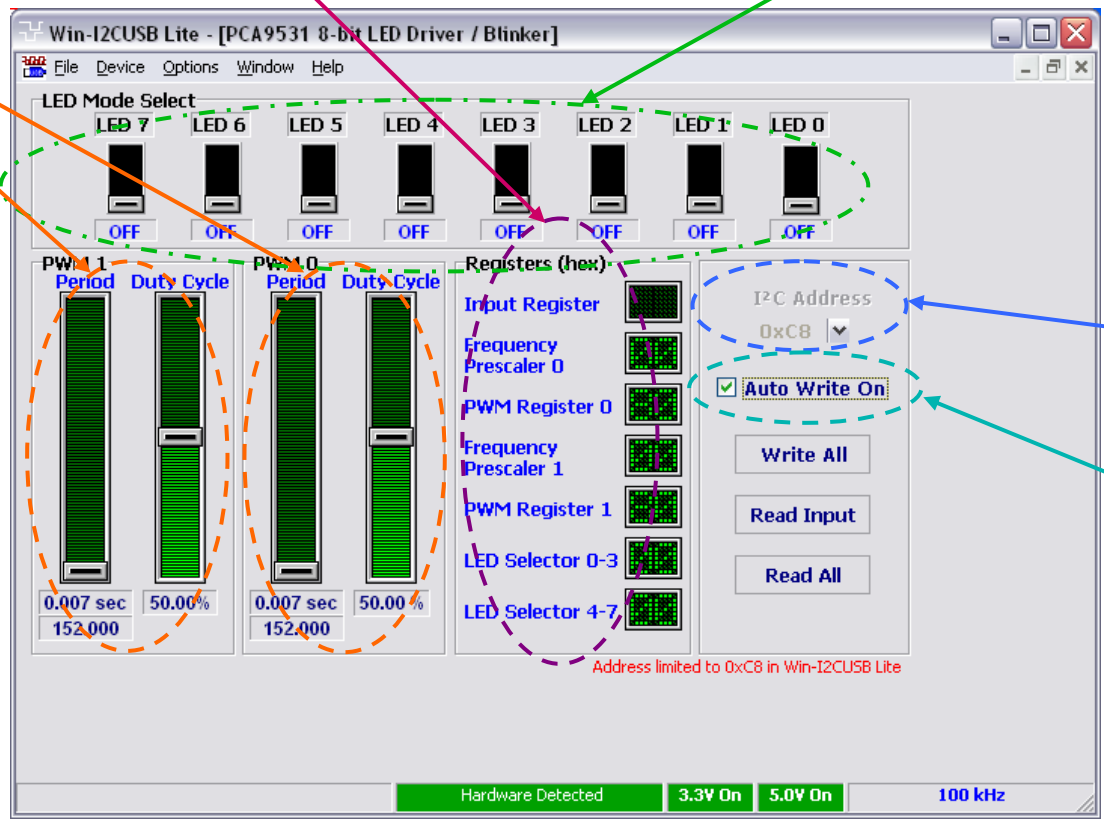
# I²C - LED Drivers, Blinkers and Dimmers

## Demoboard I2C 2005-1: PCA9531 GUI (PCA9551 similar)

Blinking / Dimming frequency and duty cycle selection

Register Hexadecimal values

LED output state selection

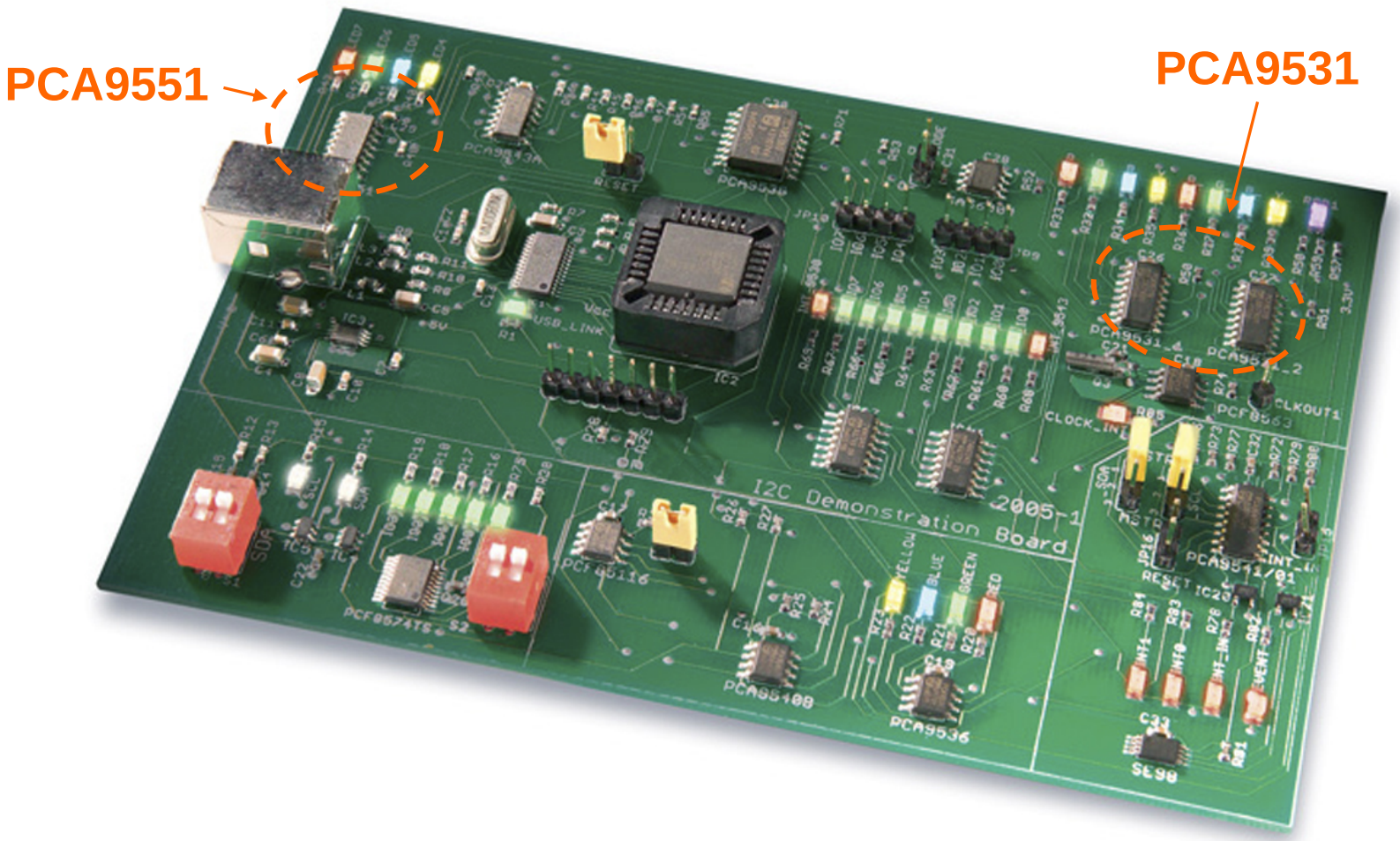



I²C address:
PCA9551 = 0xC0
PCA9531 = 0xC8

Auto Write Feature

# I²C - LED Drivers, Blinkers and Dimmers

## Demoboard I2C 2005-1: PCA9551 & PCA9531 on the board



# I²C - LED Drivers, Blinkers and Dimmers

## Demoboard I2C 2005-1: Exercise A: PCA9551

1. Power sequence the board → Board OFF then ON
2. Open the PCA9551 GUI:
   Device → LED Blinkers and Dimmers → PCA9551 8-bit LED Blinker
3. Read all the registers. What do you read?

   All the LED outputs are OFF
   Both blinking rates default to slower blinking rate (once every ~ 6.4 s, 50 % duty cycle)
4. Program the 4 LEDs connected to the PCA9551 as following:
   LED0 ( Y): ON
   LED1 ( B): Blinking Rate 0 with Blinking Rate 0 = 1s – 75 % duty cycle
   LED2 ( G): OFF
   LED3 ( R): Blinking Rate 1 with Blinking Rate 1 = 0.2 s (5 Hz) – 25 % duty cycle
   Observe the result
5. Check the Auto Write option
   Program LED0 to LED3 to Blinking Rate 0
   Use the maximum frequency for Blinking Rate 0 (40 Hz) and change the duty cycle from min to max back and forth. What do you observe?

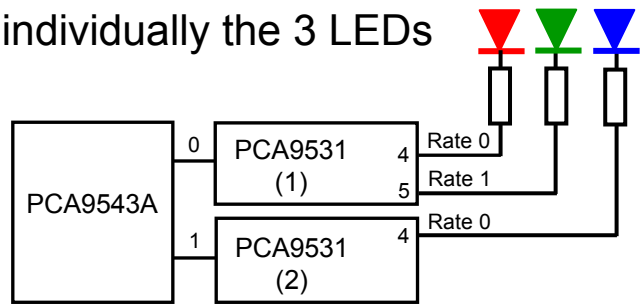
   Using the maximum frequency of the LED blinkers allow brightness control by changing the duty cycle. However, the human eye can observe some flickering (no perfect dimming) due to the fact that the dimming frequency is not high enough. Effective dimming is achieved with a PWM frequency > 80 to 100 Hz

# I²C - LED Drivers, Blinkers and Dimmers

## Demoboard I2C 2005-1: Exercise B: PCA9531

1. Power sequence the board → Board OFF then ON
2. a) Open the PCA9531 GUI:
   Device → LED Blinkers and Dimmers → PCA9531 8-bit LED Dimmer
   b) Open the PCA9543A GUI:
   Device → Multiplexers/Switches → PCA9543A 2-channel switch with Interrupt Logic

   Reminder – Architecture (see slide PCA9531 schematic):
   a) The 2 PCA9531 have the same I2C address and the PCA9543A allows to select one or the other
   (or both)
   b) The RGB LED is connected as following:
   - Red: LED4 of PCA9531 Device 1 (PCA9543A Channel 0)
   - Green: LED5 of PCA9531 Device 1 (PCA9543A Channel 0)
   - Blue: LED4 of PCA9531 Device 2 (PCA9543A Channel 1)
3. Use the PCA9543A and PCA9531 GUIs to control individually the 3 LEDs



# I²C - LED Drivers, Blinkers and Dimmers

## Demoboard I2C 2005-1: Exercise B (cont.): PCA9531

4. When it is understood how the RGB LEDs are controlled, program the two PCA9531 devices to display a “not too bright purple color” using the right amount of Red, Green and Blue
   (**Hint:** Use the PCA9543A (Switch) and PCA9531 GUIs to control individually the 3 LEDs)
5. Blink the LEDs displaying purple color
6. Now, program the two PCA9531 devices to display a “not too bright yellow color”
7. Blink the LEDs displaying Yellow color

# I²C - LED Drivers, Blinkers and Dimmers

## Demoboard I2C 2005-1: Exercise C: PCA9531, PCA9551; using the expert mode

Write the following I2C sequences (Red) using the Expert Mode and execute it step by step:

1. Connect Jumpers SDA and SCL to Master 0 (Higher position) and enable Jumper JP15 (INT_IN) → LED INT_IN should be ON
2. Power sequence the board → Board OFF then ON
3. Write to the PCA9551 to have LED0 / LED3 blinking at 1 Hz, 50% duty cycle and LED1/LED2 blinking at 2 Hz, 50 % duty cycle
4. Write to the 2 PCA9531 to have the 8 discrete blinking at 1 Hz, 50% duty cycle and the RGB LED displaying a white color
5. Run the sequence above during 2s
6. Reset the 2 PCA9531 and the PCA9551 during 2s
7. Clear the reset conditions and loop back to the 1st instruction

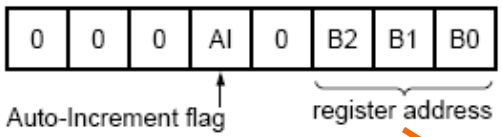
**Hints, methodology and Technical Information:**

- Use the GUI's to define the register values (easier than calculating with the formulas)
- Resets are active low
- PCA9551 Reset line controlled by I/O0 ; PCA9531 Reset line controlled by I/O4
- Control register definition: see next slide

# I²C - LED Drivers, Blinkers and Dimmers

## Demoboard I2C 2005-1: Exercise C (cont.): PCA9531, PCA9551; Control register

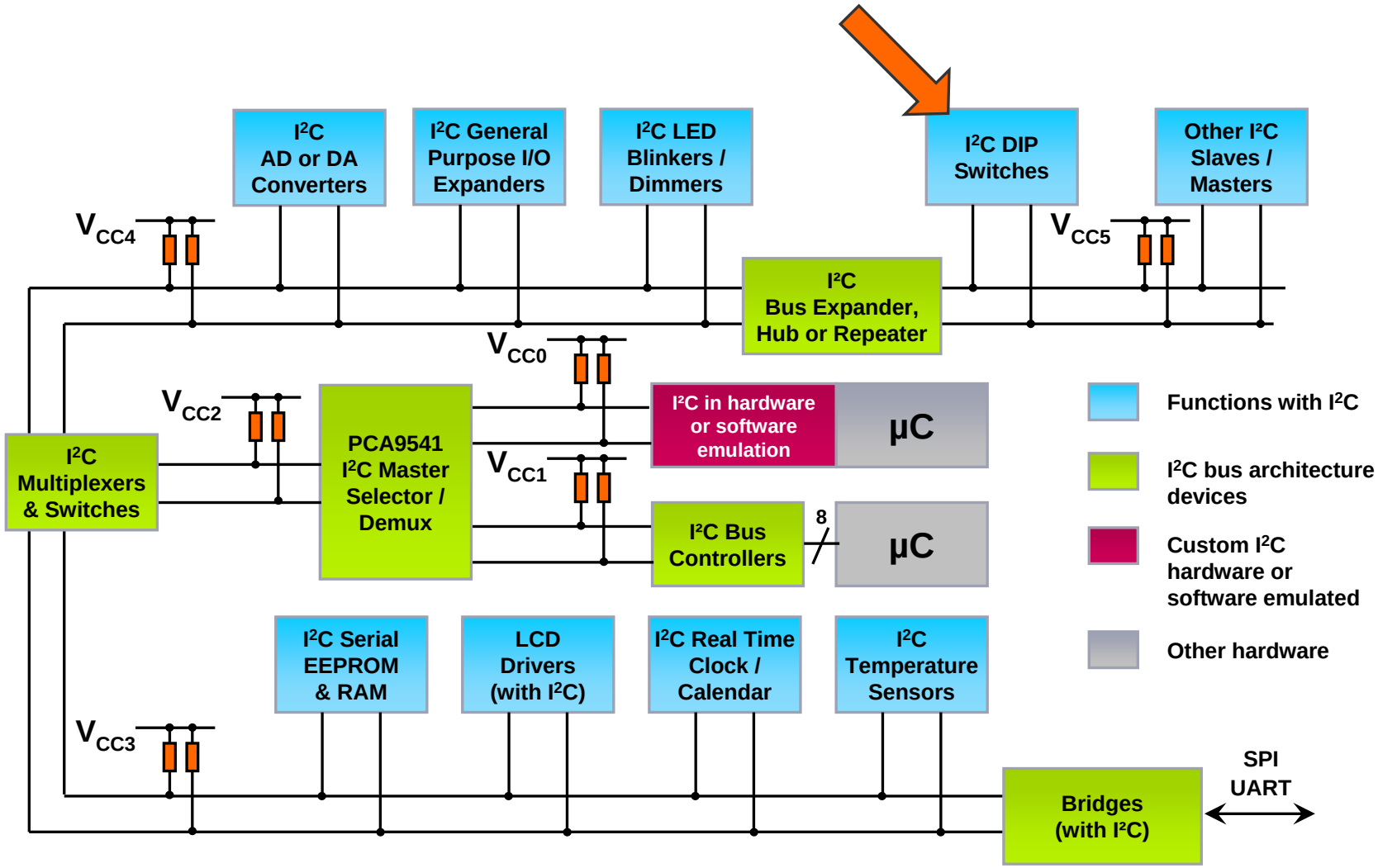
- Control register definition:




| B2 | B1 | B0 | Symbol | Access | Description |
|---|---|---|---|---|---|
| 0 | 0 | 0 | INPUT | read only | input register |
| 0 | 0 | 1 | PSC0 | read/write | frequency prescaler 0 |
| 0 | 1 | 0 | PWM0 | read/write | PWM register 0 |
| 0 | 1 | 1 | PSC1 | read/write | frequency prescaler 1 |
| 1 | 0 | 0 | PWM1 | read/write | PWM register 1 |
| 1 | 0 | 1 | LS0 | read/write | LED0 to LED3 selector |
| 1 | 1 | 0 | LS1 | read/write | LED4 to LED7 selector |

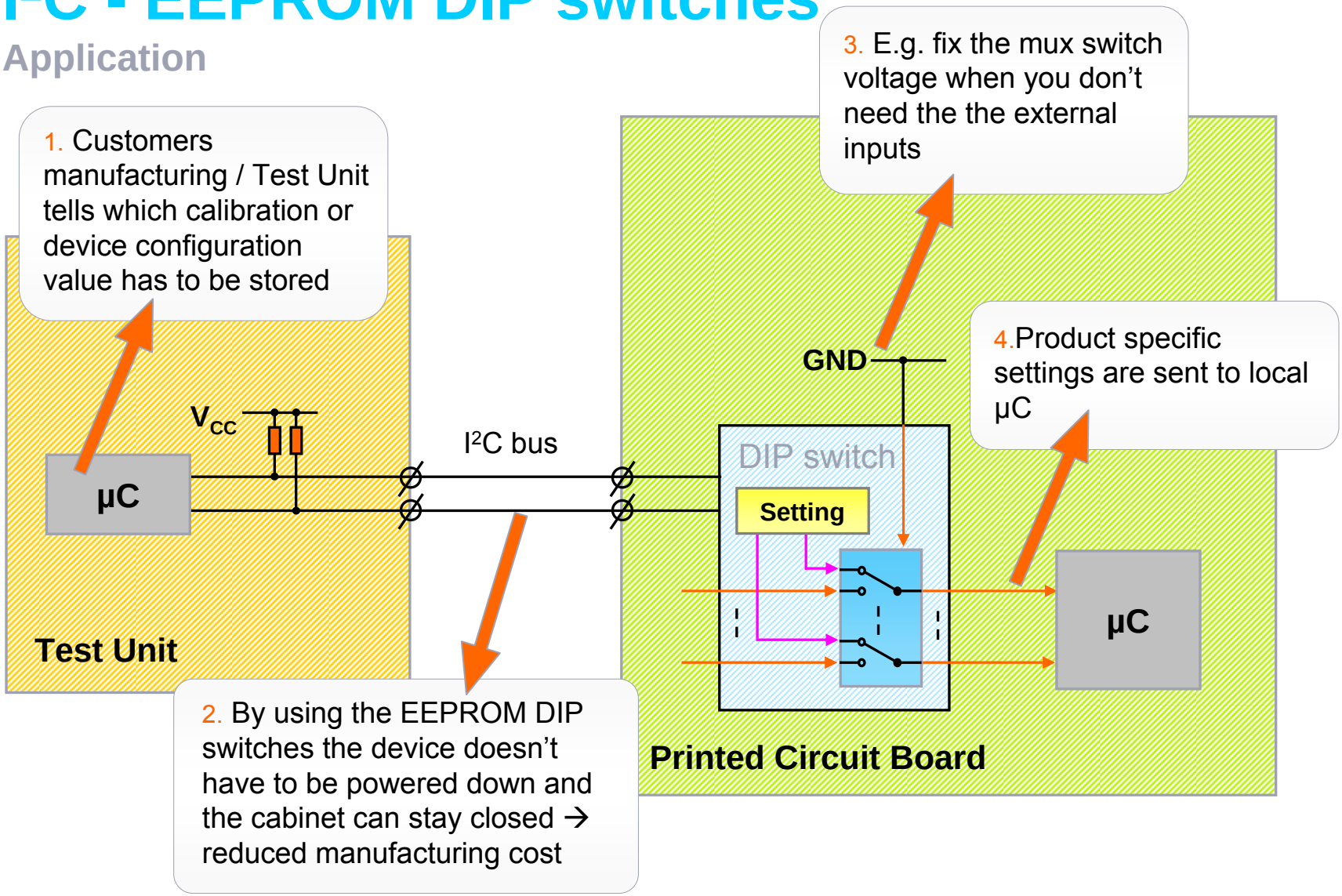
- Auto Increment feature
  - Allows more than one register to be programmed (or read) with the same command
  - Bit 4 in the Control Register (2nd byte - contains the # of the register to be addressed)
  - INPUT → PSC0 → PWM0 → PSC1 → PSC2 → LS0 → LS1 → LS2 → LS3

# I²C - EEPROM DIP switches



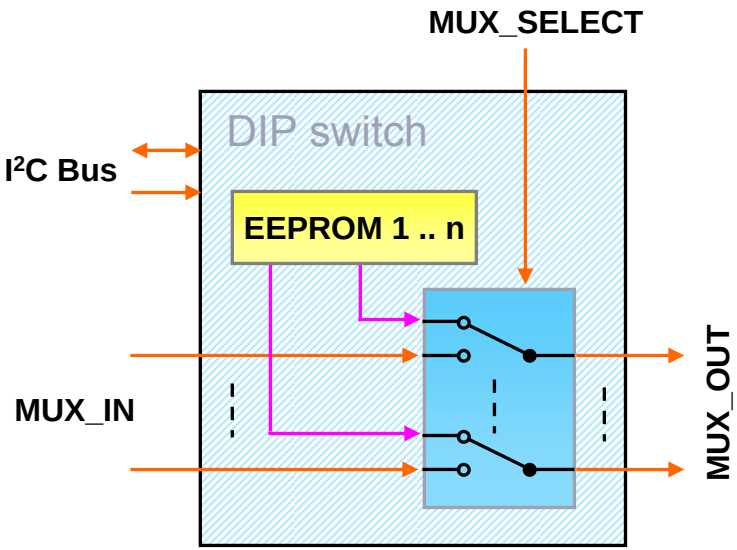
# I²C - EEPROM DIP switches

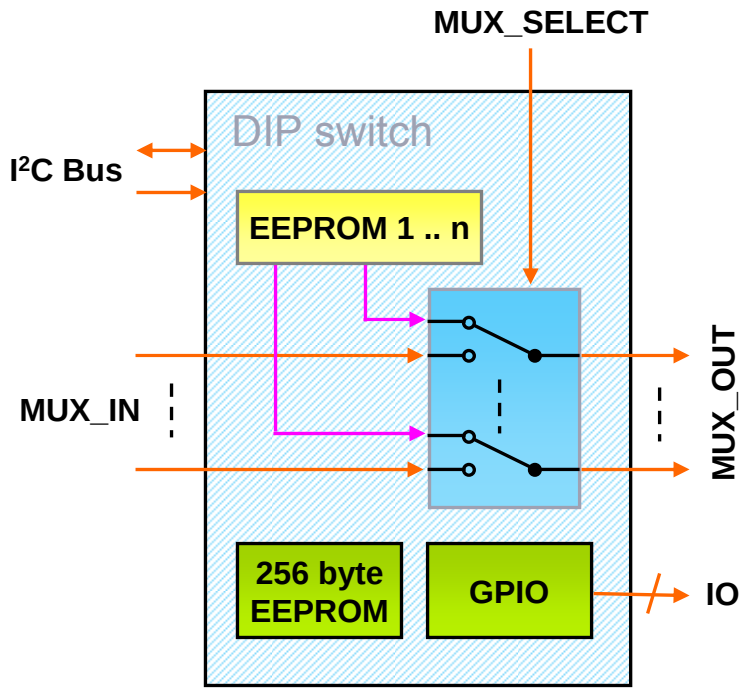
## Application



# I²C - EEPROM DIP switches

## Product overview



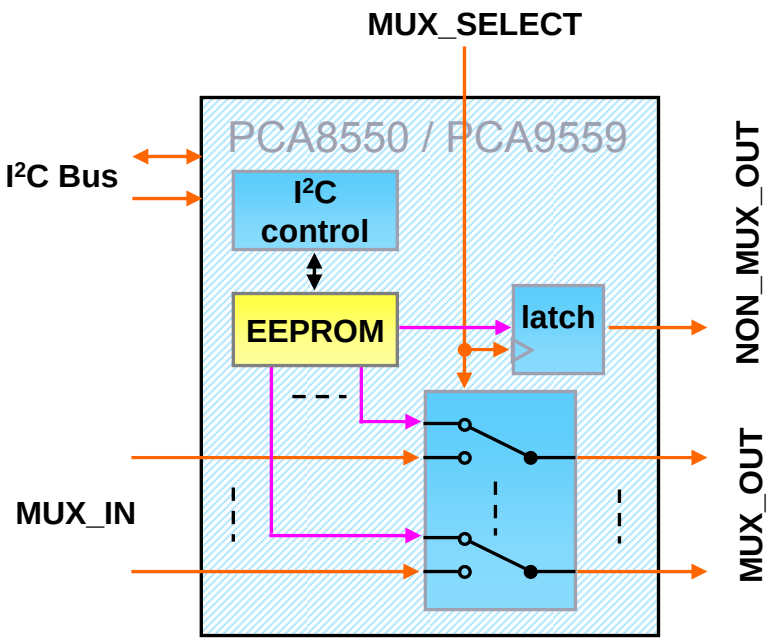



# I²C - EEPROM DIP switches

## PCA8550 / PCA9559

### General

- Useful for ‘jumperless’ configuration of PC motherboards




### PCA8550

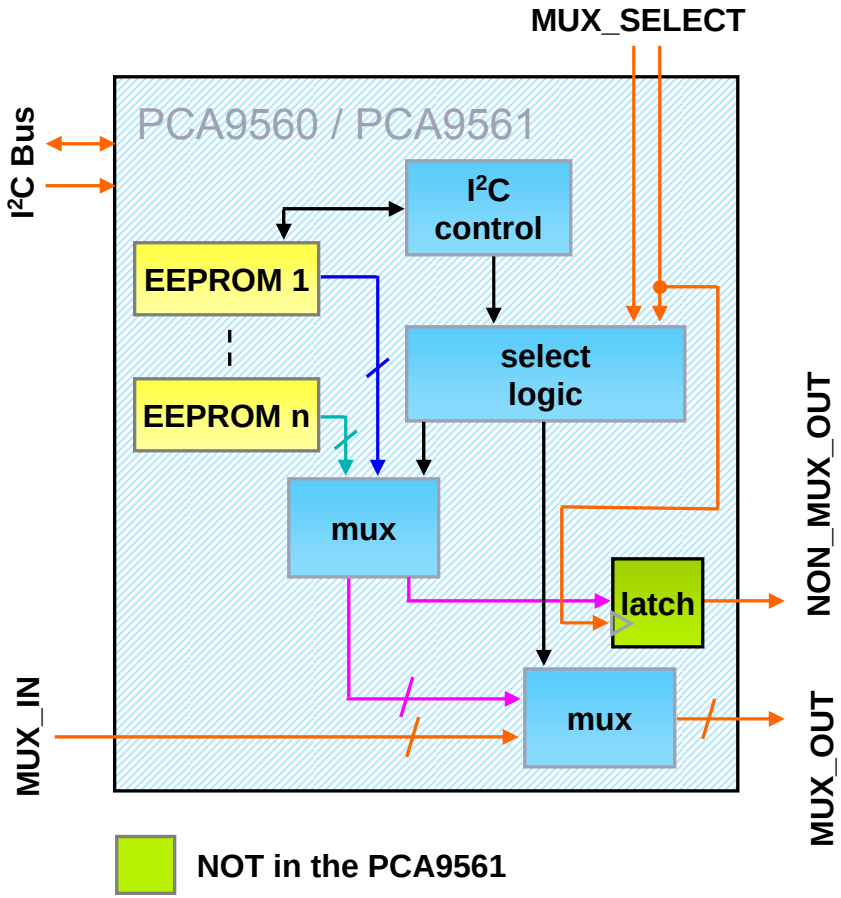
- 4-bit 2-to-1 multiplexer, 1-bit latch DIP switch
- 5-bit internal non-volatile register
- 2.5 V multiplexed outputs; 3.3 V non-multiplexed output (latched)

### PCA9559

- 5-bit 2-to-1 multiplexer, 1-bit latch DIP switch
- 6-bit internal non-volatile register
- open drain outputs
- 2 address pins, allowing up to 4 devices on the I²C-bus

# I²C - EEPROM DIP switches

## PCA9560 / PCA9561




### General

- Useful for 'jumperless' configuration of PC motherboards
- 2 address pins, allowing up to 4 devices on the I²C-bus

- PCA9560
- 5-bit 3-to-1 multiplexer, 1-bit latch DIP switch
- 5-bit external hardware pins
- Two 6-bit internal non-volatile registers, fully pin-to-pin compatible with PCA9559
- I²C/SMBus interface logic
- Open drain outputs

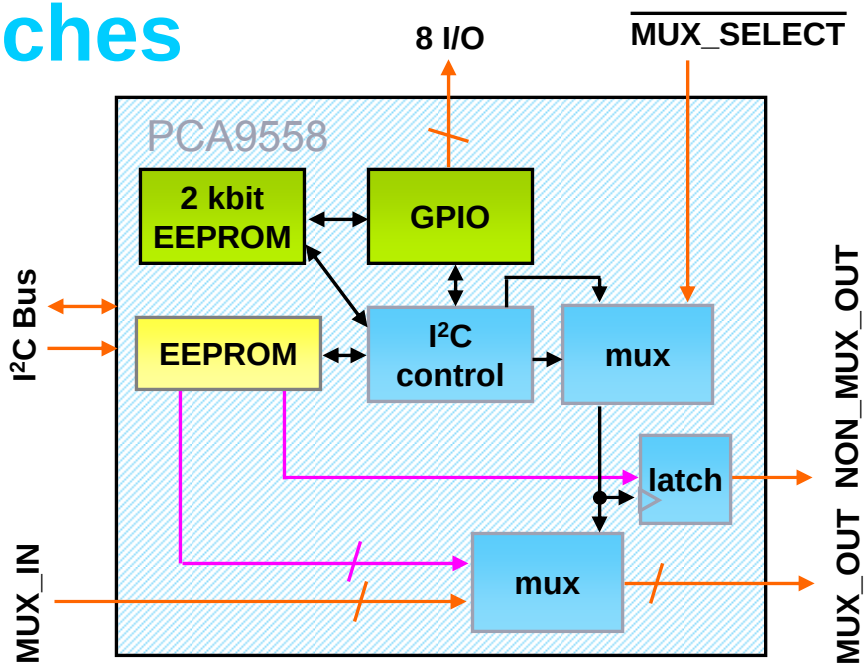
### PCA9561

- 6-bit 5-to-1 multiplexer DIP switch
- 6-bit external hardware pins
- Quad 6-bit internal non-volatile registers`

# I²C - EEPROM DIP switches

## PCA9558

- 5-bit 2-to-1 multiplexer with 1-bit latched from the 6-bit internal
- Input hardware pins readable via I²C/SMBus
- I²C and SMBus compatible
- 8-bit GPIO can directly drive LEDs
- 2 kbit (256 x 8-byte) EEPROM
- Hot insertion supported




### Application

- Multi-card systems in Telecom, Networking and Base Station
- Infrastructure Equipment

### Application (Continued)

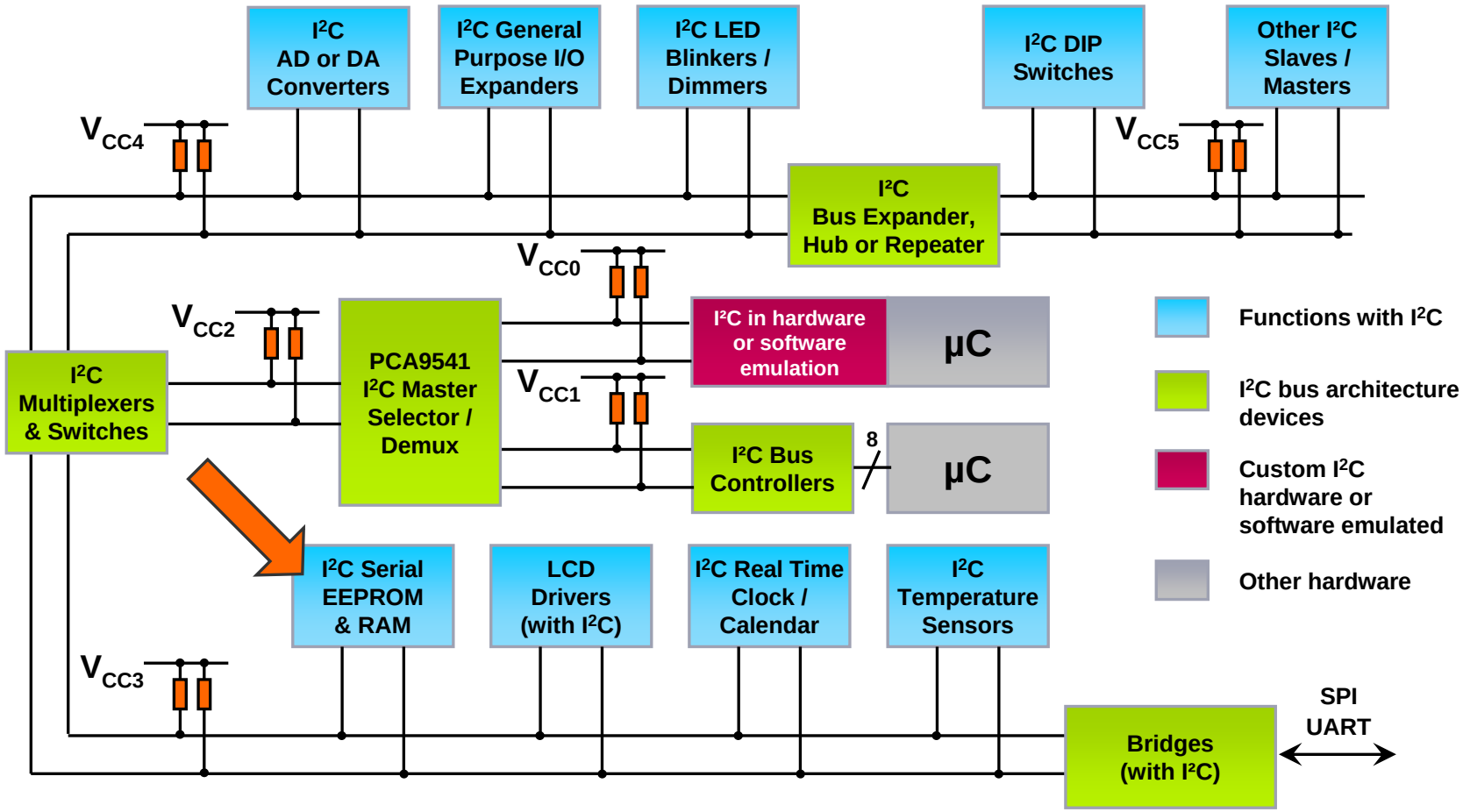
- Multi-card systems in Telecom, Networking and Base Station
- Infrastructure Equipment
- Field recall and trouble-shooting functions for installed boards
- General-purpose integrated I/O with DIP switch and memory

# I²C - EEPROM DIP switches

## Product summary

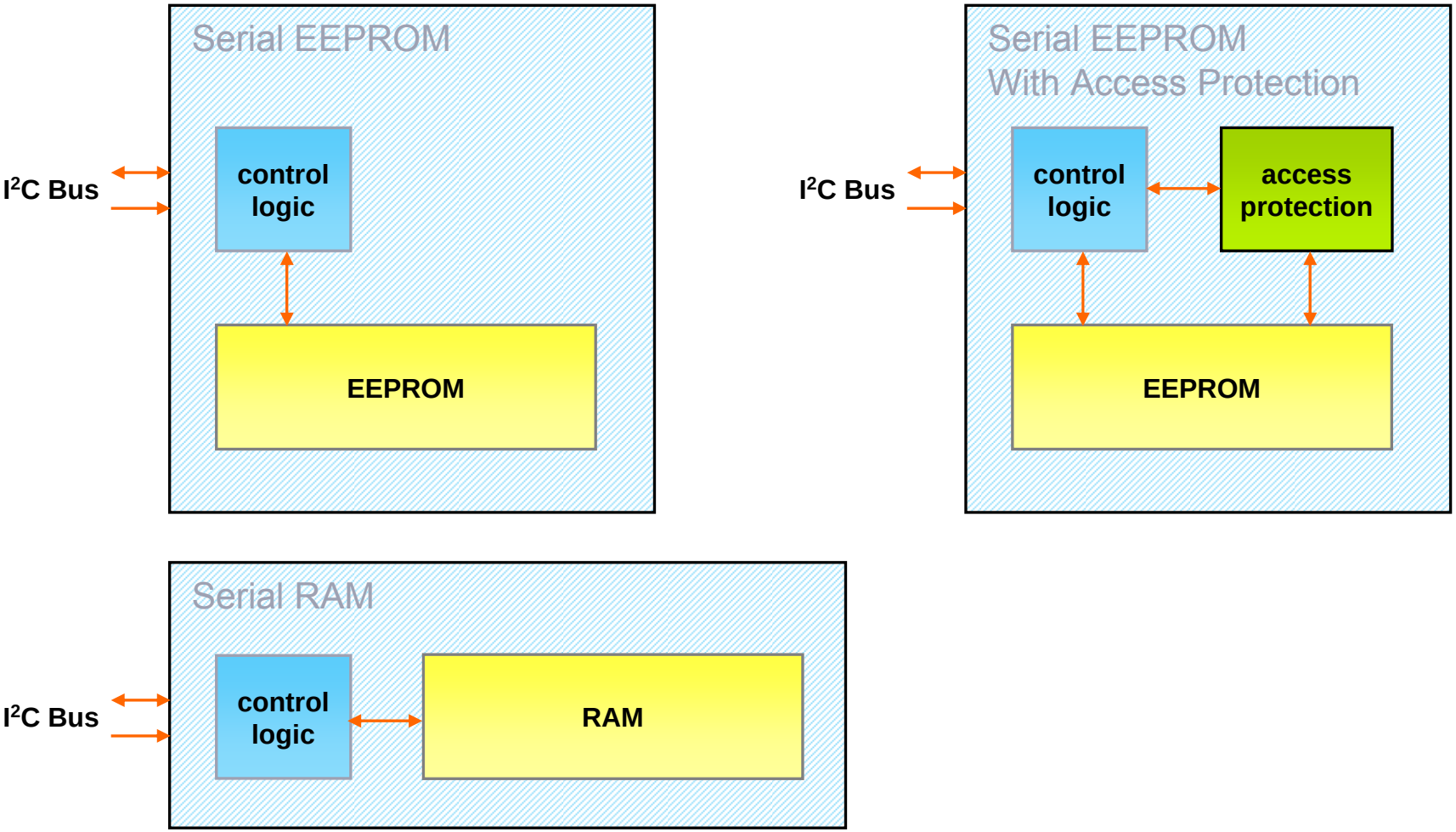
| Type nr. | # of Pins | # of Non-volatile registers | # of Register bits | # of Hardware input pins | # of Muxed outputs | Non-muxed output |
|---|---|---|---|---|---|---|
| PCA8550 | 16 | 1 | 5 | 4 | 4 | Yes |
| PCA9559 | 20 | 1 | 6 | 5 | 5 | Yes |
| PCA9560 | 20 | 2 | 6 | 5 | 5 | Yes |
| PCA9561 | 20 | 4 | 6 | 6 | 6 | No |
| PCA9558 | 28 | 1 | 6 | 5 | 5 | Yes |

# I²C - Serial EEPROM & RAM



# I²C - Serial EEPROM & RAM

## Product overview



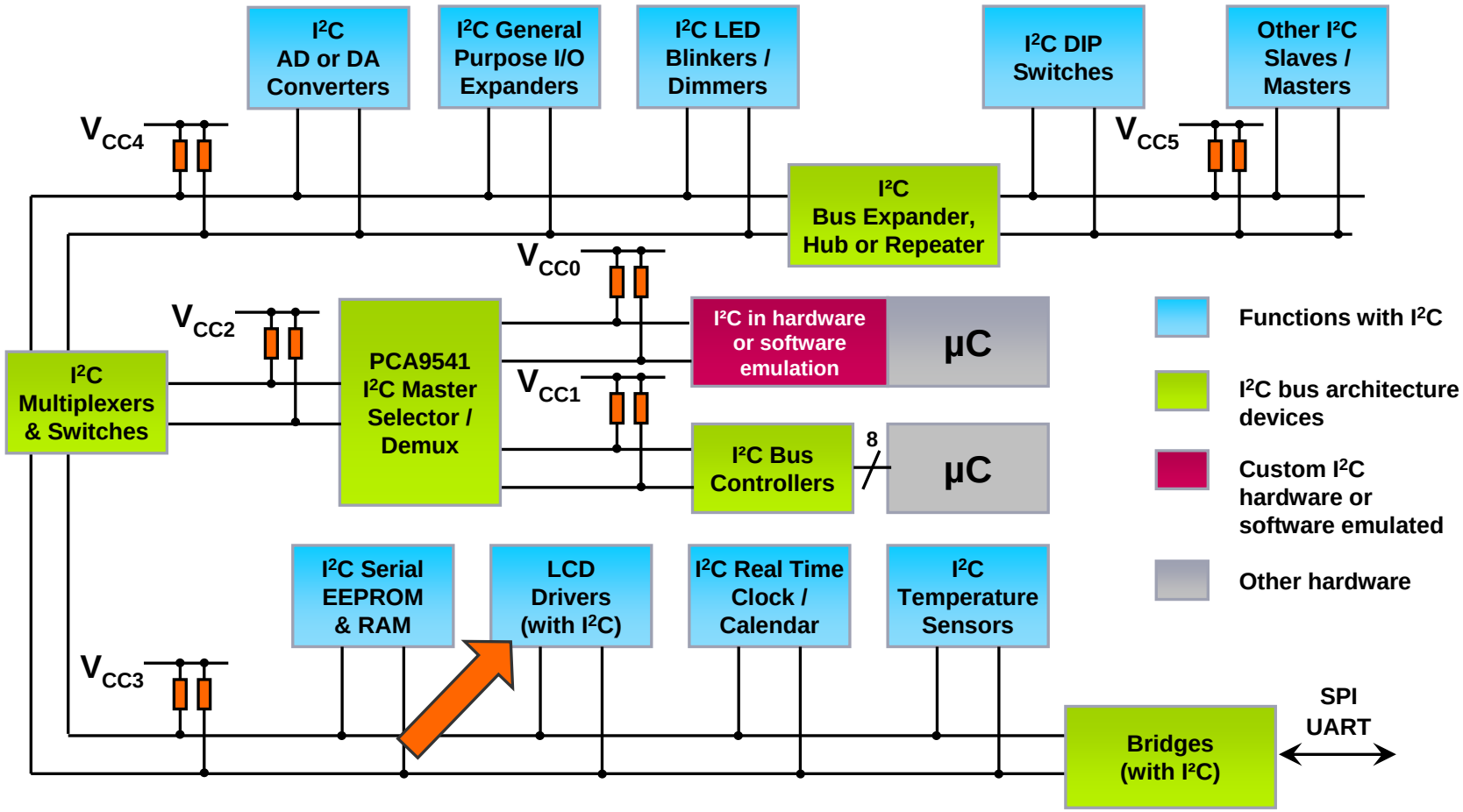
# I²C - Serial EEPROM & RAM

## Product summary

| Serial EEPROM products | power supply | Address pins | # of blocks (256 bytes) | # of bits | Data retention time | Temp range | Clock frequency |
|---|---|---|---|---|---|---|---|
| PCA8581 | 4.5 - 5.5 V | 3 | 0.5 | 1 K | 10 Years | -25 to +85°C | 100 kHz |
| PCA8581C | 2.5 - 6 V | 3 | 0.5 | 1 K | 10 Years | -25 to +85°C | 100 kHz |
| PCF8582C-2 | 2.5 - 6V | 3 | 1 | 2 K | 10 Years | -40 to +85°C | 100 kHz |
| PCF85102C-2 | 2.5 - 6 V | 3 | 1 | 2 K | 10 Years | -40 to +85°C | 100 kHz |
| PCF85103C-2 | 2.5 - 6 V | 3 | 1 | 2 K | 10 Years | -40 to +85°C | 100 kHz |
| PCF8594C-2 | 2.5 - 6 V | 2 | 2 | 4 K | 10 Years | -40 to +85°C | 100 kHz |
| PCF8598C-2 | 2.5 - 6 V | 1 | 4 | 8 K | 10 Years | -40 to +85°C | 100 kHz |
| PCF85116-3 | 2.7 - 5.5 V | 0 | 8 | 16 K | 20 Years | -40 to +85°C | 400 kHz |
| PCA24S08 | 2.5 - 3.6 V | 0 | 8(128 Bytes) | 8 K | 10 Years | -40 to +85°C | 400 kHz |

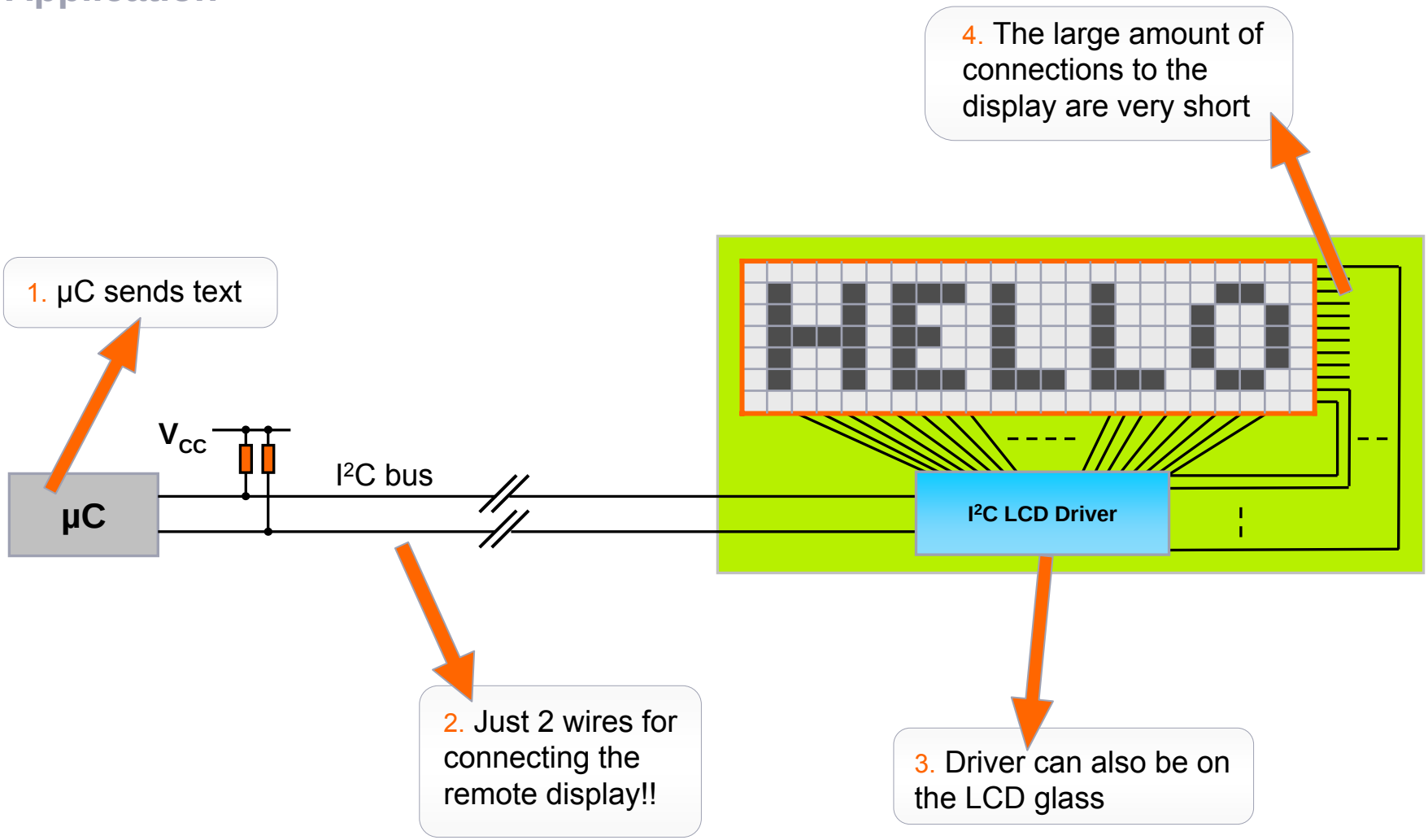
| Serial RAM products | power supply | Address pins | # of blocks (256 bytes) | # of bits | Data retention time | Temp range | Clock frequency |
|---|---|---|---|---|---|---|---|
| PCF8570 | 2.5 - 6 V | 3 | 1 | 2 K | N/A | -40 to +85°C | 100 kHz |

# I²C - LCD Drivers



# I²C - LCD Drivers

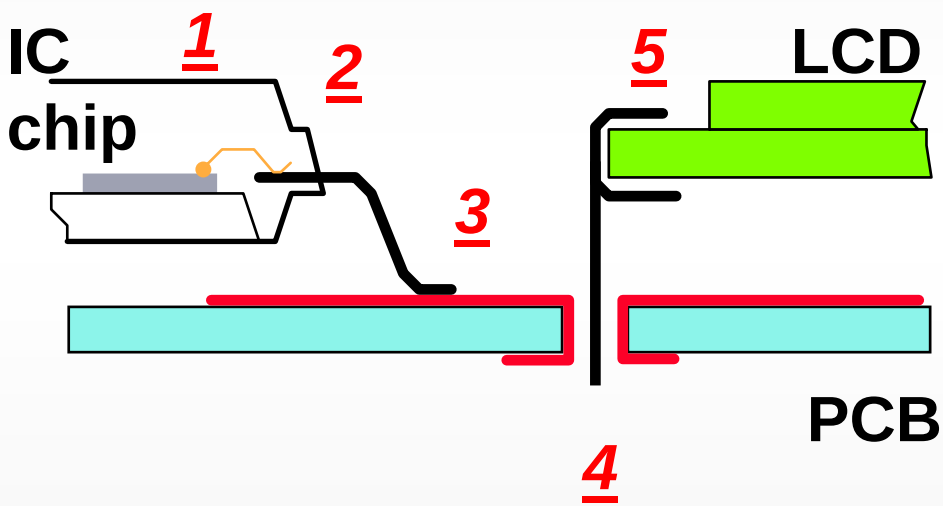
## Application



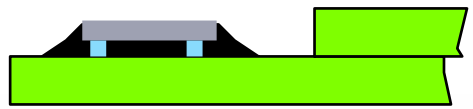
# I²C - LCD Drivers

## Principle of Chip On Glass (COG)

### Conventional



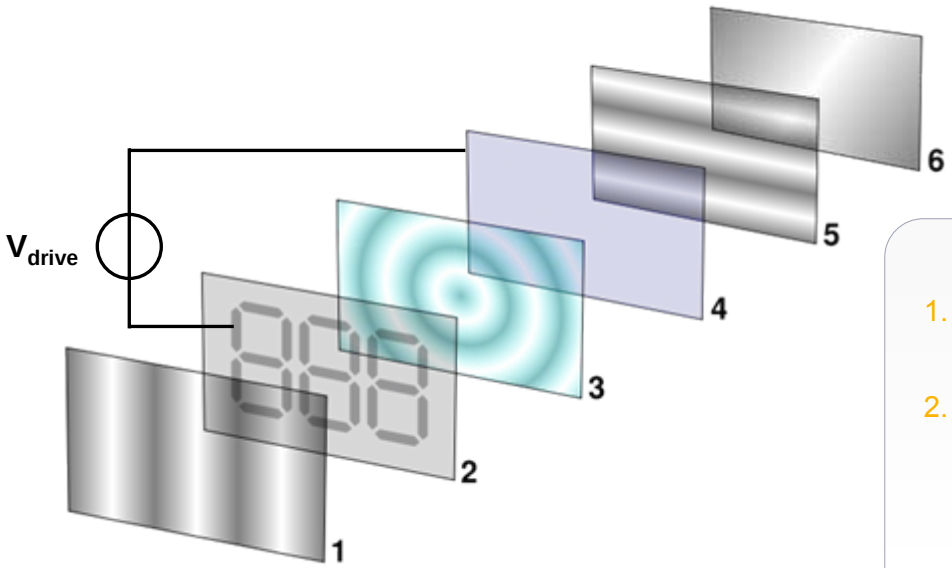

### Chip On Glass



***only 1***

# I²C - LCD Drivers

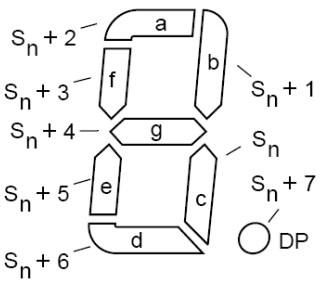
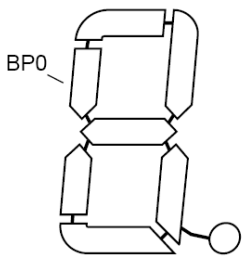
## Intermezzo: LCD basics




1. Vertical filter film to polarize the light as it enters.
2. Glass substrate with ITO electrodes. The shapes of these electrodes will determine the dark shapes that will appear when the LCD is turned on or off. Vertical ridges etched on the surface are smooth.
3. Twisted nematic liquid crystals
4. Glass substrate with common electrode film (ITO) with horizontal ridges to line up with the horizontal filter (backplane).
5. Horizontal filter film to block/allow through light.
6. Reflective surface to send light back to viewer.

# I²C - LCD Drivers

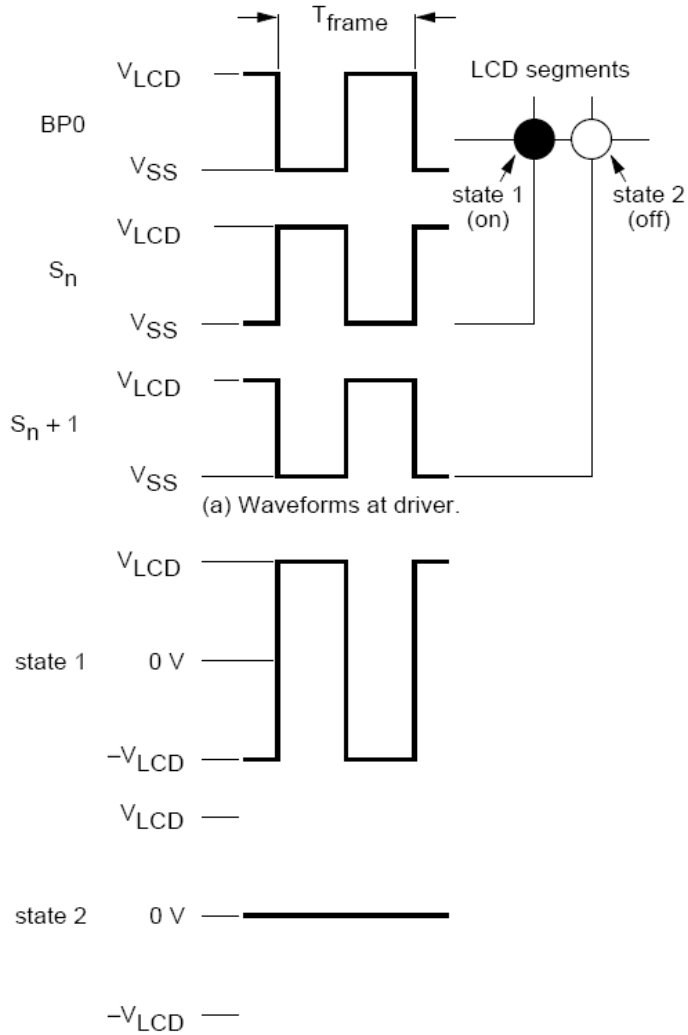
## Intermezzo: driving LCD : static

| drive mode | LCD segments | LCD backplanes |
|---|---|---|
| static |  |  |



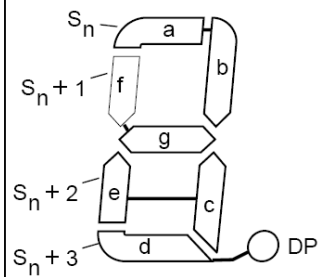
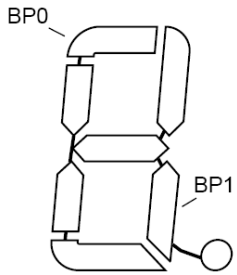


- LCD displays should be driven AC. (Prolonged DC operation will cause reduced life)
- Typically drive frequency: between 30 Hz and 100 Hz

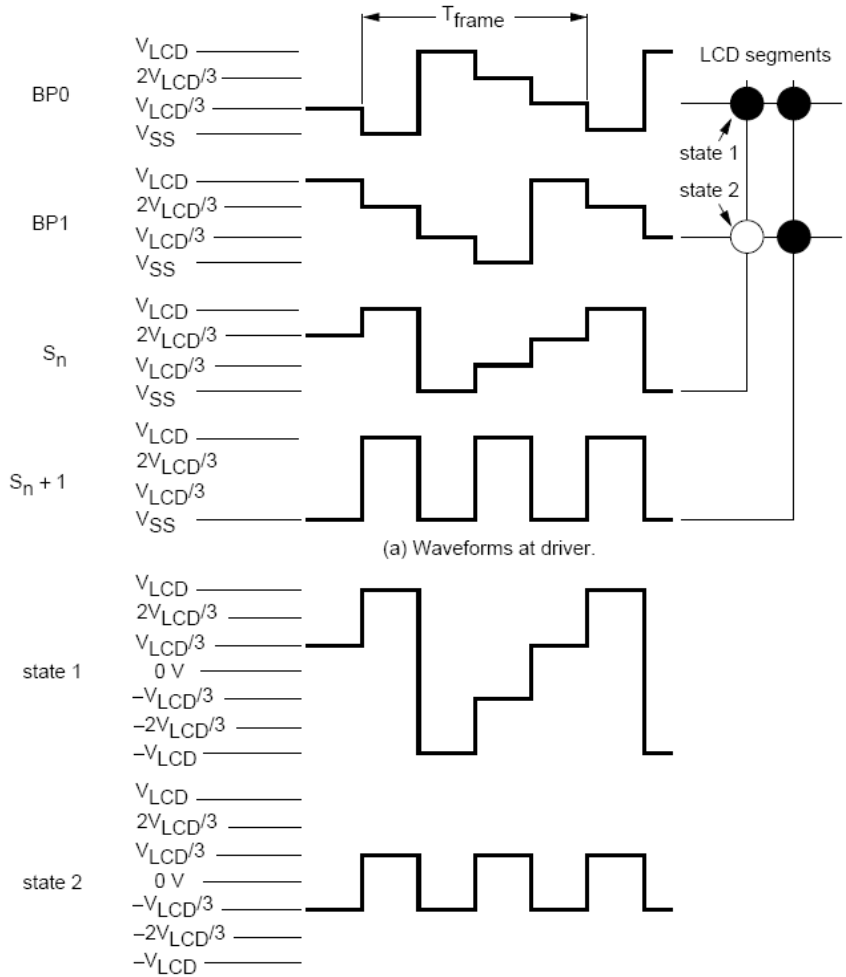


# I²C - LCD Drivers

## Intermezzo: driving LCD: multiplex 1:2

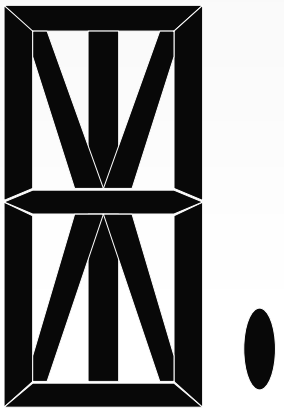
| drive mode | LCD segments | LCD backplanes |
|---|---|---|
| 1 : 2<br><br>multiplex |  |  |

- Multiplex: reduce number of drive pins
- Reduction of external connections:
  - Enhances device reliability
  - Increases the potential display density.
- Multiple voltage biasing levels are needed

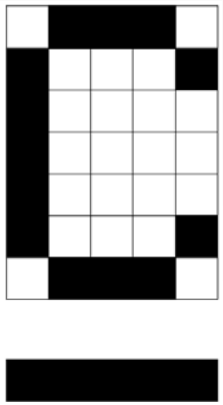


# I²C - LCD Drivers

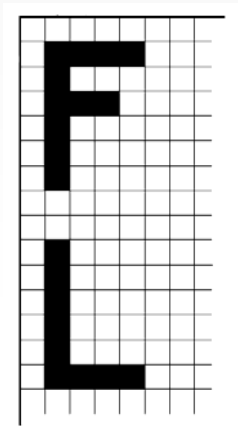
## Families




**Segment:**
drive electrode; can be any shape



**Character:**
column / row (pixel) driven; character set in memory of driver



**Graphic:**
column / row driven (pixel)

# I²C - LCD Drivers

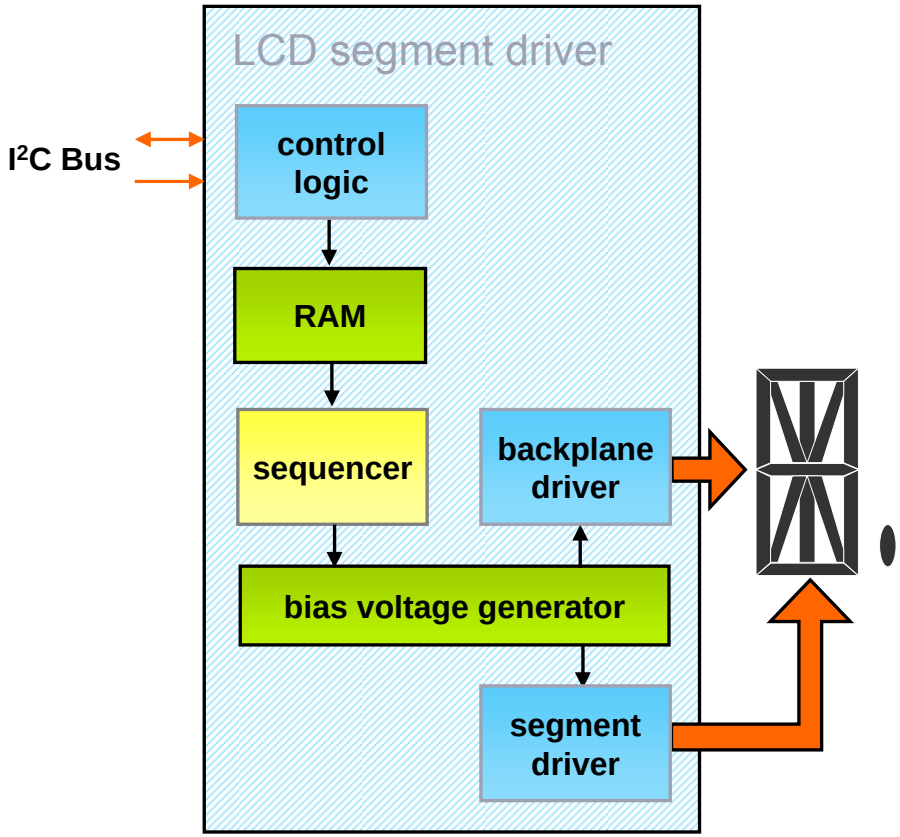
## LCD segment drivers

Key products

- PCF8562 4 x 32 segments 8 μA
- PCF8576D 4 x 40 segments 8 μA
- PCF8534 4 x 60 segments 24 μA
- PCF8533 4 x 80 segments 24 μA

Features

- Wide range of segment outputs
- On-chip RAM with auto incremental addressing
- Low power consumption
- No external components
- Wide power supply range
- On-chip LCD bias voltage generation



# I²C - LCD Drivers

## Product summary - segment drivers

| Type | multiplex rates versus number of segments | | | | | | Interface bus | Package | | Expandable | $V_{dd}$ [V] | $V_{LCD}$ [V] | $I_{tot}$ [µA] | Comments |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1:1 | 1:2 | 1:3 | 1:4 | 1:8 | 1:16 | | | | | | | | |
| OM4068H | 32 | 64 | 96 | | | | SPI | QFP-44 | | OM4068 | 2.5 - 5.5 | 3.5 - 6.5 | 65 (max) | Good for SPI based systems |
| PCF8533 | 80 | 160 | 240 | 320 | | | I²C | | | PCF8533 up to 5120 segments | 1.8 - 5.5 | 2.5 - 6.5 | 32 (80 max) | |
| PCF8534AH | 60 | 120 | 180 | 240 | | | I²C | LQFP-80 | | PCF8534 up to 480 segments | 1.8 - 5.5 | 2.5 - 6.5 | 32 (80 max) | High segment density, lower power |
| PCF8566T | 24 | 48 | 72 | 96 | | | I²C | VSO-40 | * | PCF8576, PCF8566 up to 1536 segments | 2.5 - 6.0 | 2.5 - 6.0 | 30 (90 max) | High runner in automotive, fitness & medical equipement |
| PCF8577C | 32 | 64 | | | | | I²C | | | PCF8577C up to 256 segments | 2.5 - 6.0 | 2.5 - 6.0 | 25 (75 max) | |
| PCF2100CT | 20 | 40 | | | | | C/Serial | SO-28 | | | 2.25 - 6.0 | 2.25 - 6.0 | 20 (50 max) | Great for small displays |
| PCF2111CT | 32 | 64 | | | | | C/Serial | VSO-40 | | | 2.25 - 6.0 | 2.25 - 6.0 | 20 (50 max) | Easy to use serial interface |
| PCF2112CT | 32 | | | | | | C | VSO-40 | | PCF2112C | 2.25 - 6.0 | 2.25 - 6.0 | 20 (50 max) | Easy to use serial interface |
| PCF8562TT/2 | 32 | 64 | 96 | 128 | | | I²C | TSSOP-48 | | | 1.8 - 5.5 | 2.5 - 6.5 | 32 (80 max) | Great for new designs. Lower power and thin package |
| PCF8576T | 40 | 80 | 120 | 160 | | | I²C | VSO-56 | ** | PCF8576, PCF8566 up to 2560 segments | 2.0 - 9.0 | 2.0 - 9.0 | 180 (max) | Good for low temperature out door applications |
| PCF8576CTT | 40 | 80 | 120 | 160 | | | I²C | HTSSOP56 | ** | PCF8576C, PCF8566 up to 2560 segments | 2.0 - 6.0 | 2.0 - 6.0 | 120 (max) | Smaller package than CT, good for board update |
| PCF8576DT/2 | 40 | 80 | 120 | 160 | | | I²C | TSSOP-56 | | PCF8576D up to 2560 segments | 1.8 - 5.5 | 2.5 - 6.5 | 32 (80 max) | Great for new designs. Lower power and thin package |
| PCF8578T | - | - | - | - | 256 | 384 | I²C | VSO-56 | ** | PCF8579 up to 40960 dots | 2.5 - 6.0 | 2.5 - 9.0 | 20 (50 max) | Scalable for dot matrix or high segment count displays |
| PCF8579T | segment extention together with PCF8578 | | | | | | I²C | VSO-56 | ** | PCF8578 up to 40960 dots | 2.5 - 6.0 | 2.5 - 9.0 | 9 (20 max) | Use up to 32 devices for a 40k dot display |

* Also as bare die for COB
** Also as bare die for COB & COG

# I²C - LCD Drivers

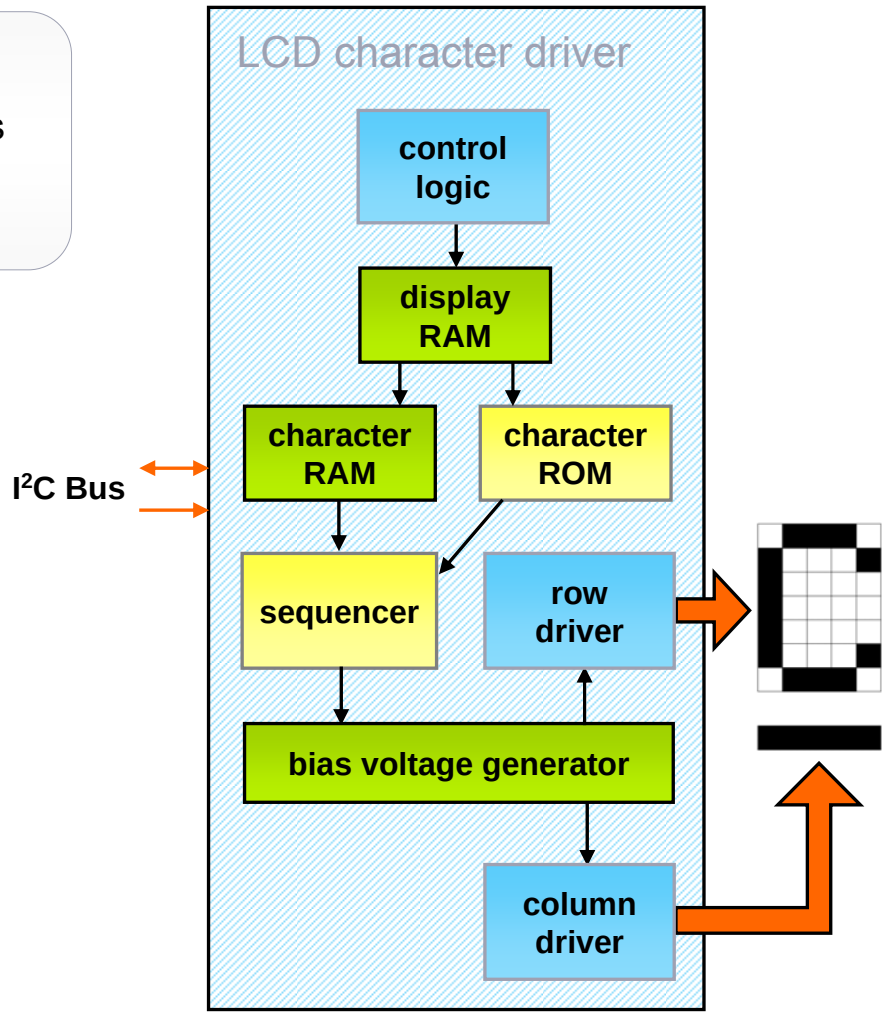
## LCD character drivers

Key products

- PCF2113 2 line by 12 characters + 120 icons
- PCF2119 2 line by 16 characters + 160 icons

Features

- On-chip character generator
- On-chip temperature compensation
- On-chip character ROM and RAM
- Low power consumption
- Minimum of external components
- On-chip LCD bias voltage generation
- Cursor support



# I²C - LCD Drivers

## Product summary – character drivers

| Type | # of characters [lines] x [char], # icons | Interface bus | # char sets (type) | $V_{LCD}$ [V] | Internal $V_{LCD}$ generation | $I_{tot}$ [µA] |
|---|---|---|---|---|---|---|
| PCF2113 | 2 x 12 or 1 x 24, 120 | I2C, 4/8 bit parallel | 1 (on request) | 2.2 - 6.5 | programmable temperature compensation | 200 |
| PCF2119 | 2 x 16, 160 | I2C, 4/8 bit parallel | 1 (on request) | 2.2 - 6.5 | programmable temperature compensation | 190 |

note: 240 characters in ROM, 16 characters in RAM; character size: 5 x 8 dots or 5 x 7 dots + cursor

# I²C - LCD Drivers

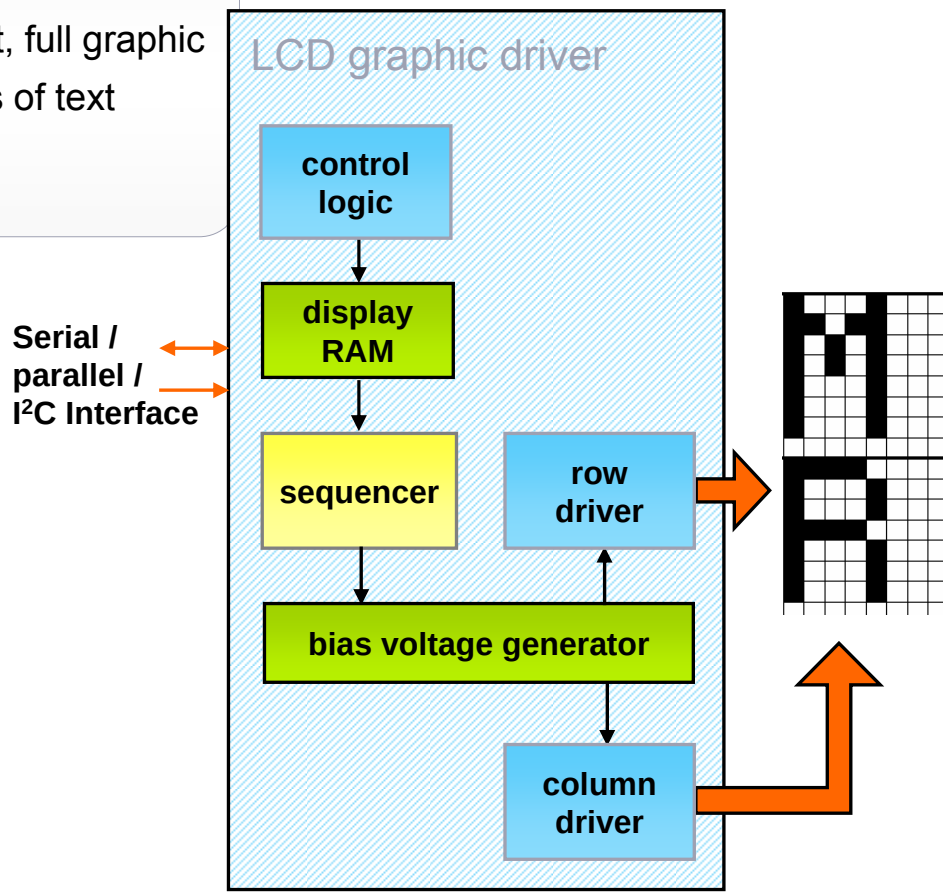
## LCD graphic drivers

Key products

- PCF8531 34 x 128 small 4 x 20 chars of text, full graphic
- PCF8535 65 x 133 medium graphics, 8 lines of text
- PCF8811 80 x 128 large universal display

Features

- Wide range of mux rates to optimize power and display size
- On-chip generation of LCD bias voltages
- Low number of external components
- Low power consumption
- Temperature compensation
- PCF8535:
  - Temperature read back
  - Icon mode



# I²C - LCD Drivers

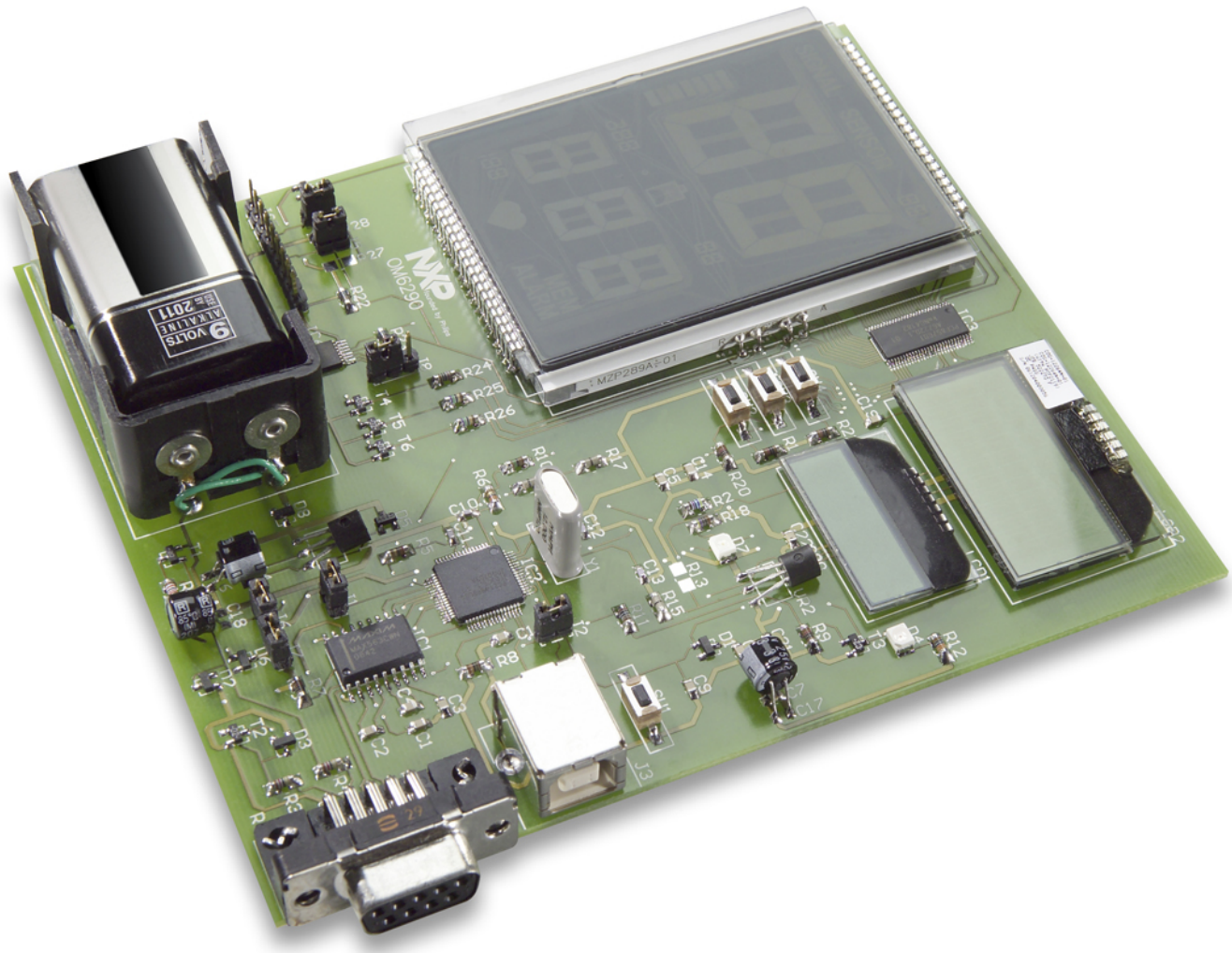
## Product summary – graphic drivers

| Type | Display | Mux rates | Interface bus I²C SPI | Package | $V_{dd}$ [V] | $V_{LCD}$ [V] | $I_{dd}$ [µA] |
|---|---|---|---|---|---|---|---|
| PCF8531 | 34 x 128 | 1: 34, 26, 17 | 400 kbps | U (die w/ bumps) | 1.5 - 2.5 | 3.0 - 9.0 | 170 |
| PCF8535 | 65 x 133 | 1: 65, 49, 26, 17 | 400 kbps | U (die w/ bumps) | 4.5 - 5.5 | 8.0 - 16 | 160 |
| PCF8811 | 80 x 128 | 1: 80, .., 16 (steps of 8) | 400 kbps<br>9 Mbps<br>(+ parallel 6800) | U (die w/ bumps) | 1.8 - 3.3 | 3.0 - 9.0 | 150 |

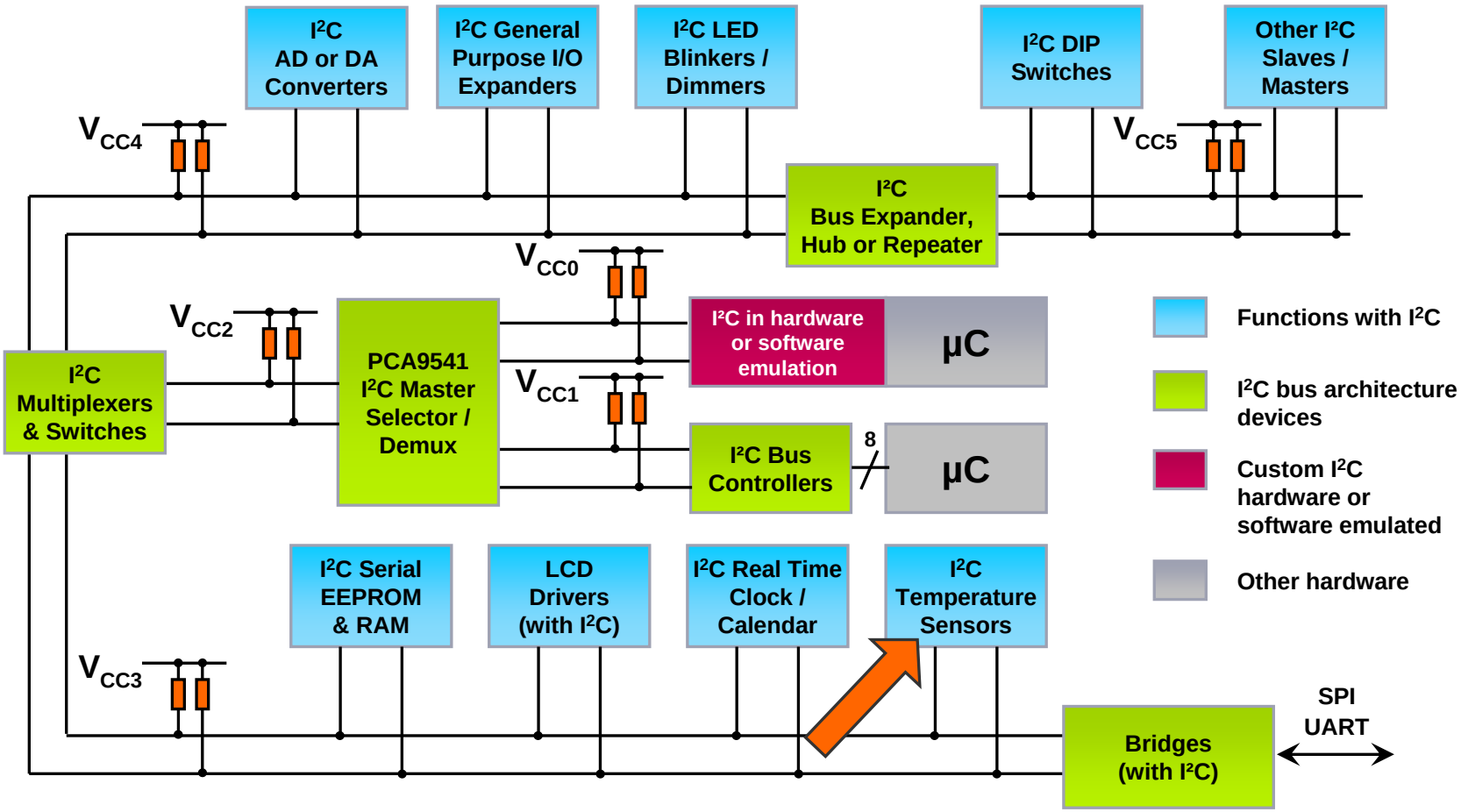
Note: Graphic display with Chip-on-glass LCD driver is a space and cost optimal solution.

# I²C - LCD Drivers

## Demoboard - OM6290

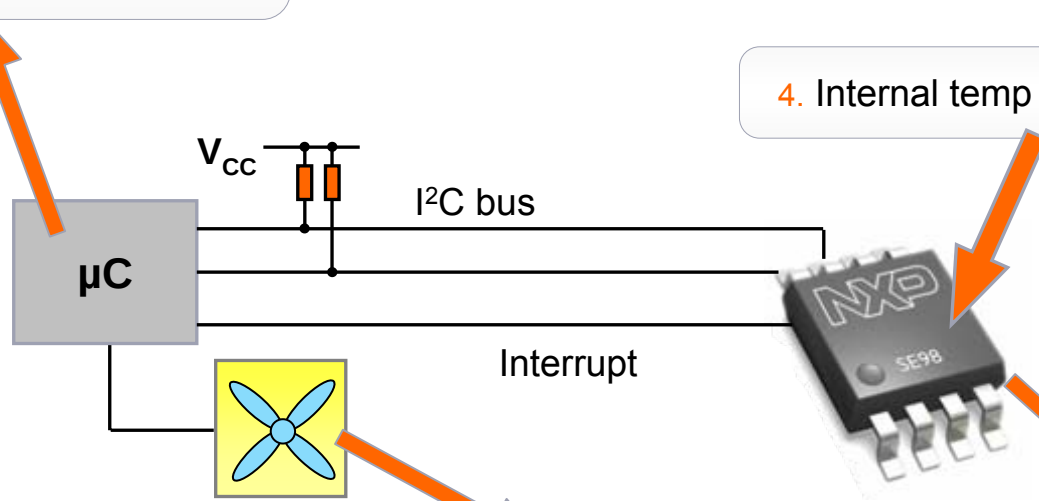


# I²C - Temperature and Voltage Sensors



# I²C - Temperature and Voltage Sensors

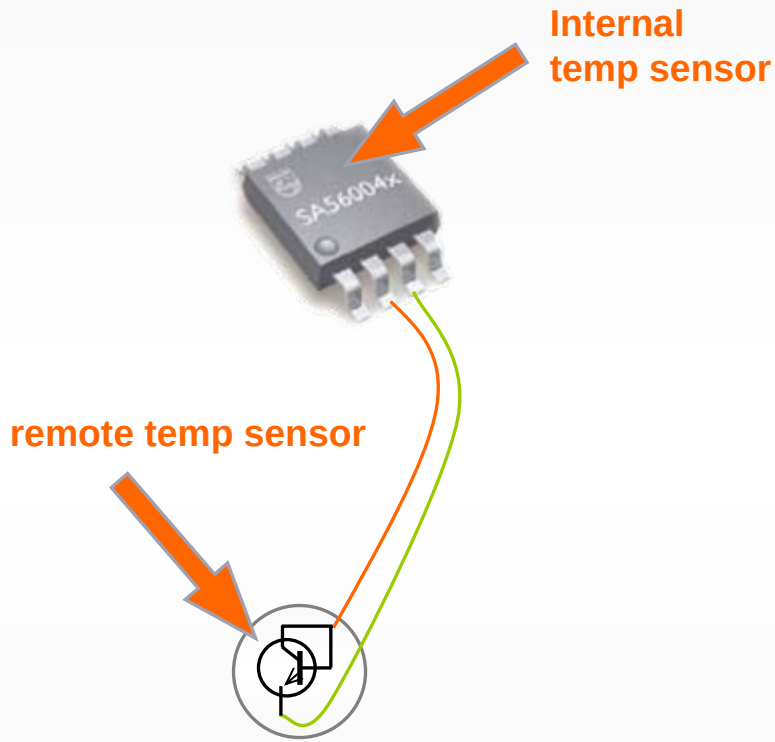
## Application



# I²C - Temperature and Voltage Sensors

## Families

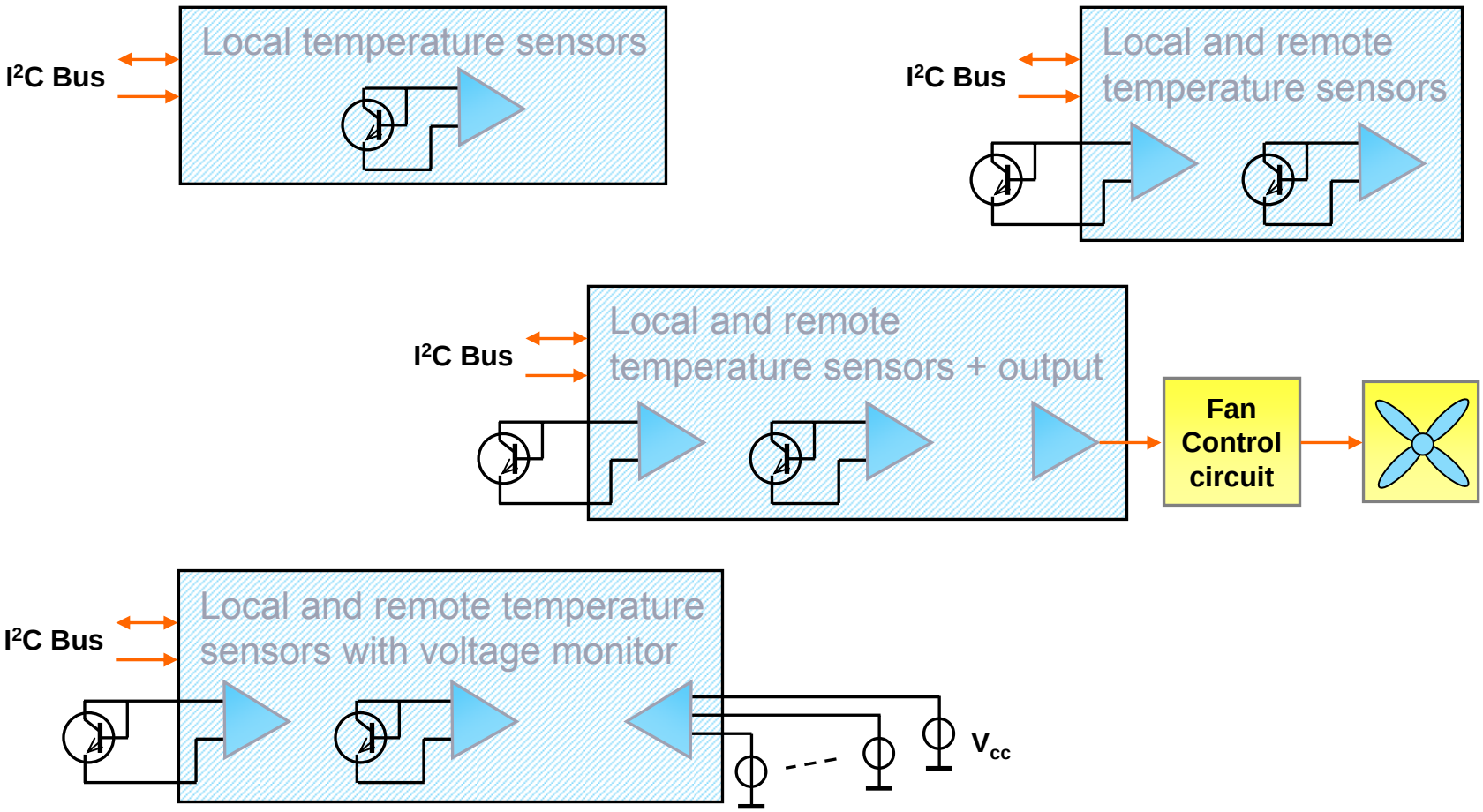

Local / internal temp sensor




Local / internal AND remote temp sensor

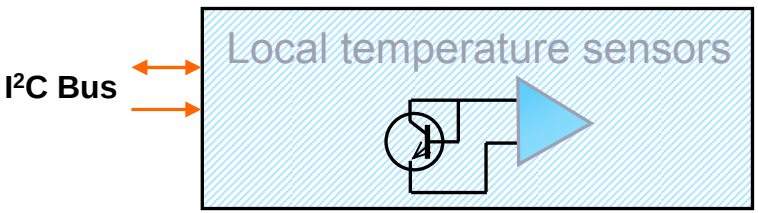
# I²C - Temperature and Voltage Sensors

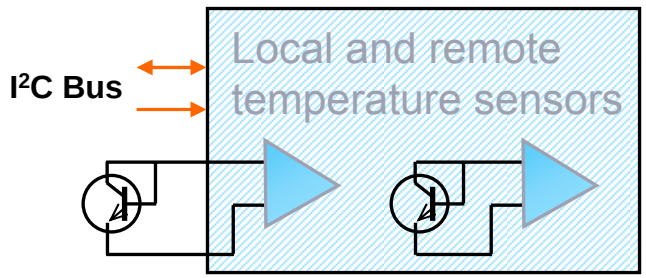
## Product overview



# I²C - Temperature and Voltage Sensors

## Product overview



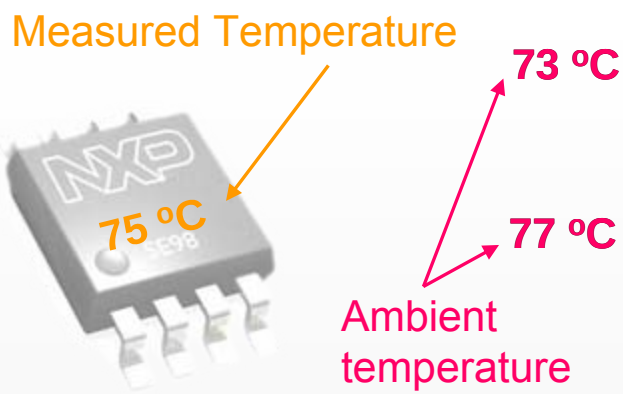



# I²C - Temperature and Voltage Sensors

## Intermezzo: Accuracy versus Resolution

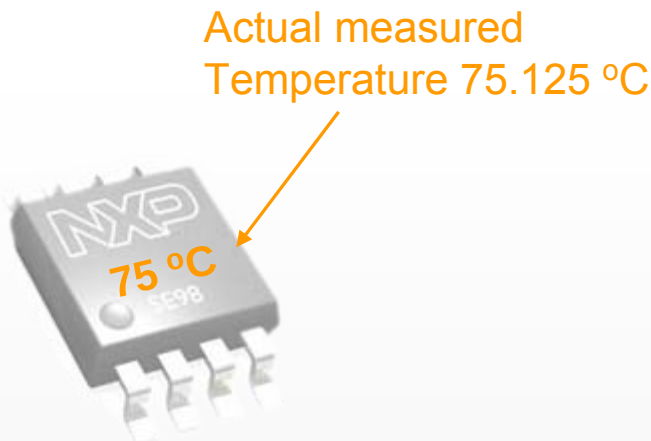
### Temperature Accuracy

- Accuracy and error are used synonymously, and is the measure of how precisely the temperature sensor reading matches the ambient temperature
- Example: ± 2 °C accuracy



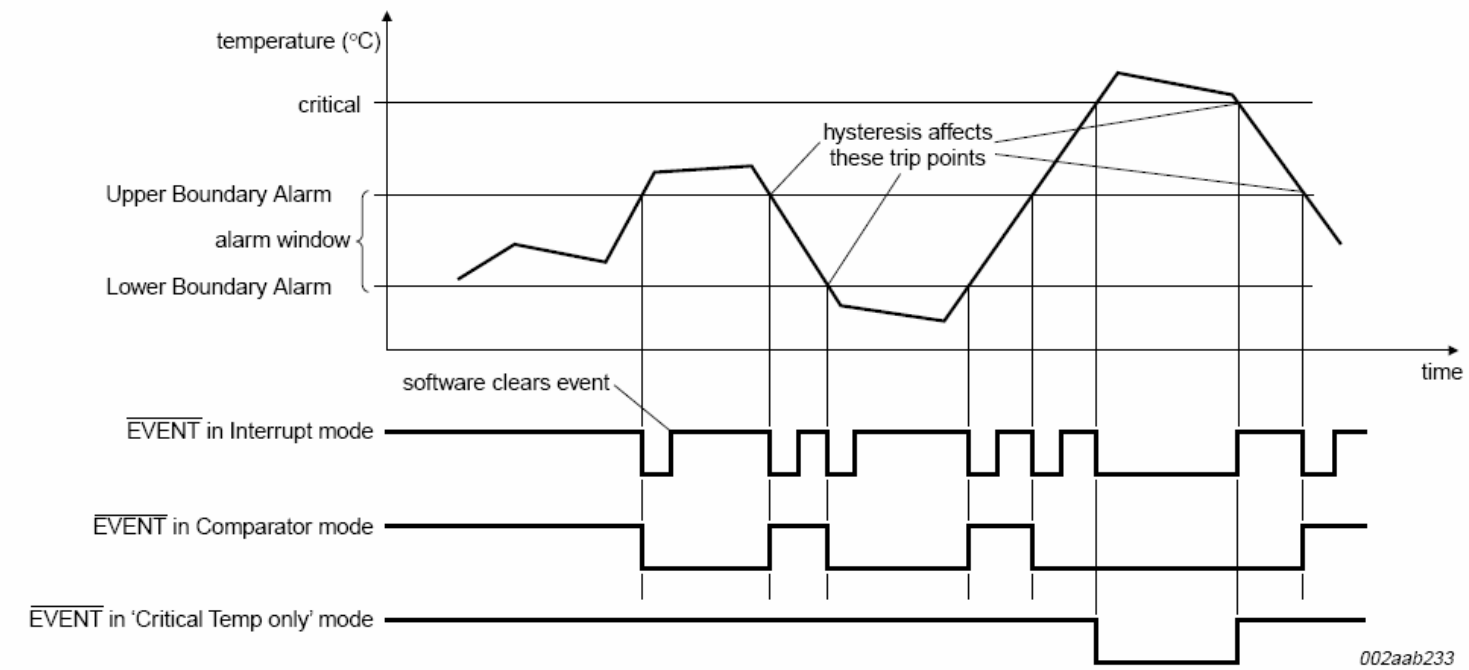

### Temperature Resolution

- Is the measure of temperature sensor's smallest measuring step
- Example: 0.125 °C resolution



# I²C - Temperature and Voltage Sensors

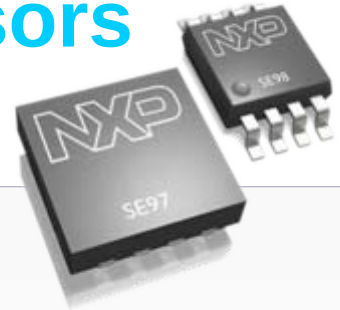
## Intermezzo: Comparator versus Interrupt mode




- Comparator mode: EVENT output is cleared by itself once the temperature drops below the set points
- Interrupt mode: EVENT output is latched and only cleared by writing to the clear event bit

# I²C - Temperature and Voltage Sensors

## Local-only temperature sensors




General:

- Produce highly accurate digital readings of the ambient temperature
- Read out via I²C or SMBus

Application:

- Trigger interrupts, shut-down, or over-temperature alarms
- Industrial process control
- Notebook computers
- Servers
- Memory modules
- Communications
- Office electronics

Products:

- *LM75A/LM75B*:
  - Local temperature sensor and watchdog timer™
  - Accuracy of ±2 ºC
  - Drop-in replacement LM75
- *SE95*:
  - More accurate version of the LM75A
  - Delivers superior performance in power-sensitive applications
- *SE98*:
  - Designed for applications that use DDR3 DIMM memory **(not only DIMM applications.)**
  - Complies with JEDEC JC42.4
  - Supports SMBus Timeout and Alert
  - Security lock bits
- *SE97*:
  - Single device: SE98 + 2-Kbit EEPROM Serial Presence Detect (SPD)

# I²C - Temperature and Voltage Sensors

## LM75A versus LM75B

- LM75B is an improved features and drop-in replacement for LM75A

LM75A and LM75B Comparison

| Features | LM75A | LM75B |
|---|---|---|
| Single byte temperature register read without lockup | No | Yes |
| Fault time-out (minimum 75 ms) | No | Yes |
| Minimum I²C-bus hold time (tHD;DAT) | 10 ns | 0 ns |
| 2 mm × 3 mm HUSON8 package | No | Yes |

# I²C - Temperature and Voltage Sensors

## Remote and local temperature sensors


General:

- Produce highly accurate digital readings of the ambient temperature
- Remotely monitor the temperature of the thermal diode inside the CPU or the diode connected to PNP or NPN transistors
- To save power in laptop applications, the standby pin (STBY) can be tied to the battery's "suspend" output.

Application:

- Trigger interrupts, shut-down, or over-temperature alarms
- Industrial process control
- Notebook computers
- Servers
- Office electronics

Products:

- *NE1617A*:
  - Up to nine slave devices on same bus
  - Drop-in replacement NE1617, AD1021 or AD1021A
- *NE1619*:
  - Integrated voltage monitor: track five input supply voltages in the range of 0 to 12 V with accuracy of ±2%
- *SA56004*:
  - Handheld and portable applications
  - Includes offset register for system calibration
  - Dual outputs for fan control
  - Interrupt
  - Built-in diode fault detection
  - One-shot conversion with power optimization in shutdown mode
  - Eight possible pre-configured slave device addresses
  - Small, 8-pin package

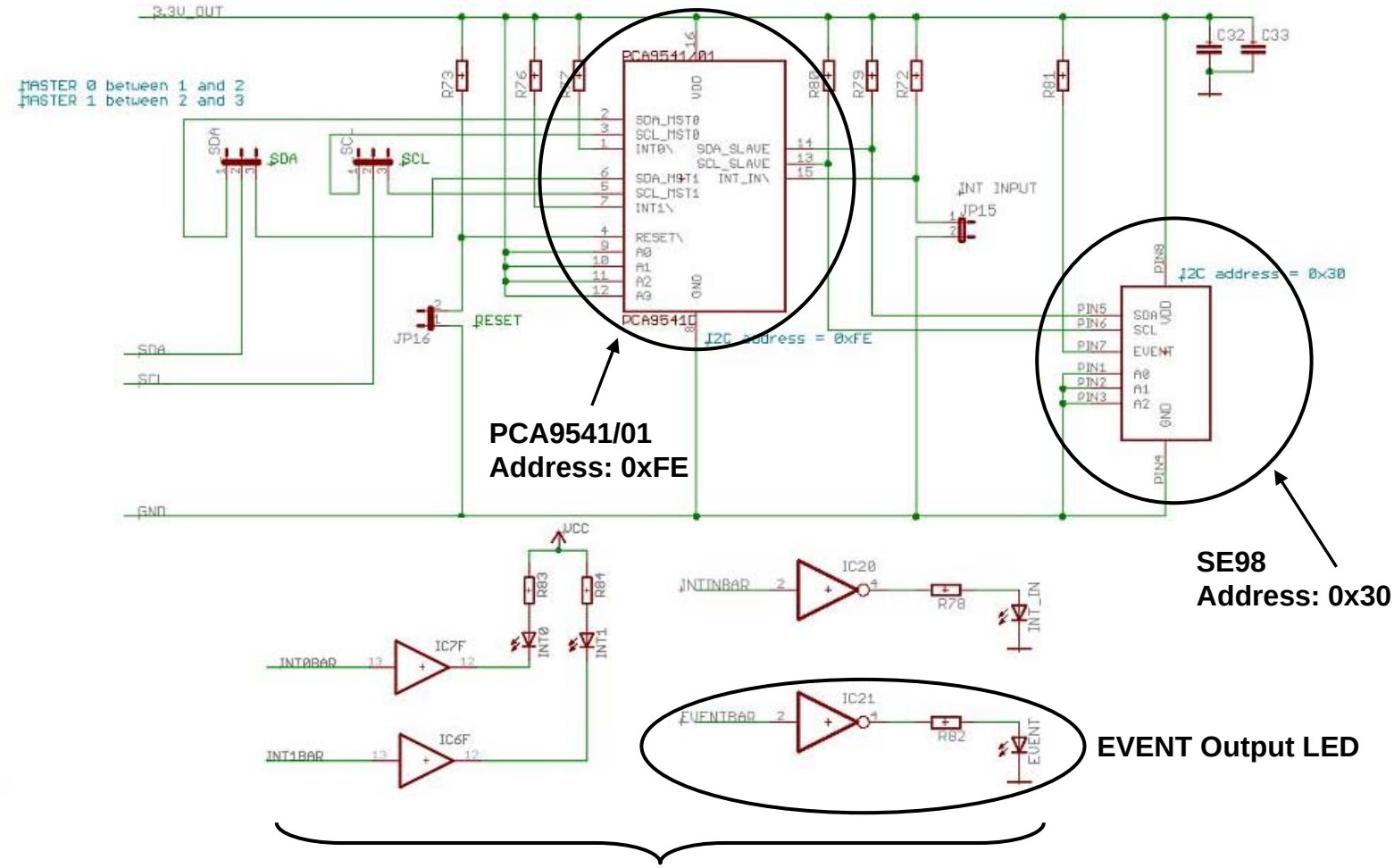
# I²C - Temperature and Voltage Sensors

## Product summary

| Type nr. | Local channels | Remote channels | Thermal Alarm Output | Fan control output | 0 .. 12 V voltage input monitor | Accuracy Local Sensing | Accuracy Remote Sensing | A/D resolution (°C # bits) | Supply range | Operating current (µA) | Power down current (µA) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LM75A | 1 | | 1 | | | ±2°C | | 0.125/11 | 2.8-5.5 | 1000 | 3.5 |
| NE1617A | 1 | 1 | 1 | | | ±2°C | ±3°C | 1.0/8 | 3.0-5.5 | 70 | 3 |
| NE1619 | 1 | 1 | | | 5 | ±3°C | ±5°C | 1.0/8 | 2.8-5.5 | 500 | 100 |
| SA56004 | 1 | 1 | 1 | 1 | | ±2°C | ±1°C | 0.125/11 | 3.0-3.6 | 500 | 10 |
| SE95 | 1 | | 1 | | | ±1°C | | 0.03125/13 | 2.8-5.5 | 1000 | 7.5 |
| SE97/97A | 1 | | 1 | | | ±1°C | | 0.125/11 | 3.0-3.6 | 250 | 15 |
| SE98 | 1 | | 1 | | | ±2°C | | 0.125/11 | 3.0-3.6 | 250 | 15 |
| SE98A | 1 | | 1 | | | ±1°C | | 0.125/11 | 1.7-3.6 | 250 | 15 |

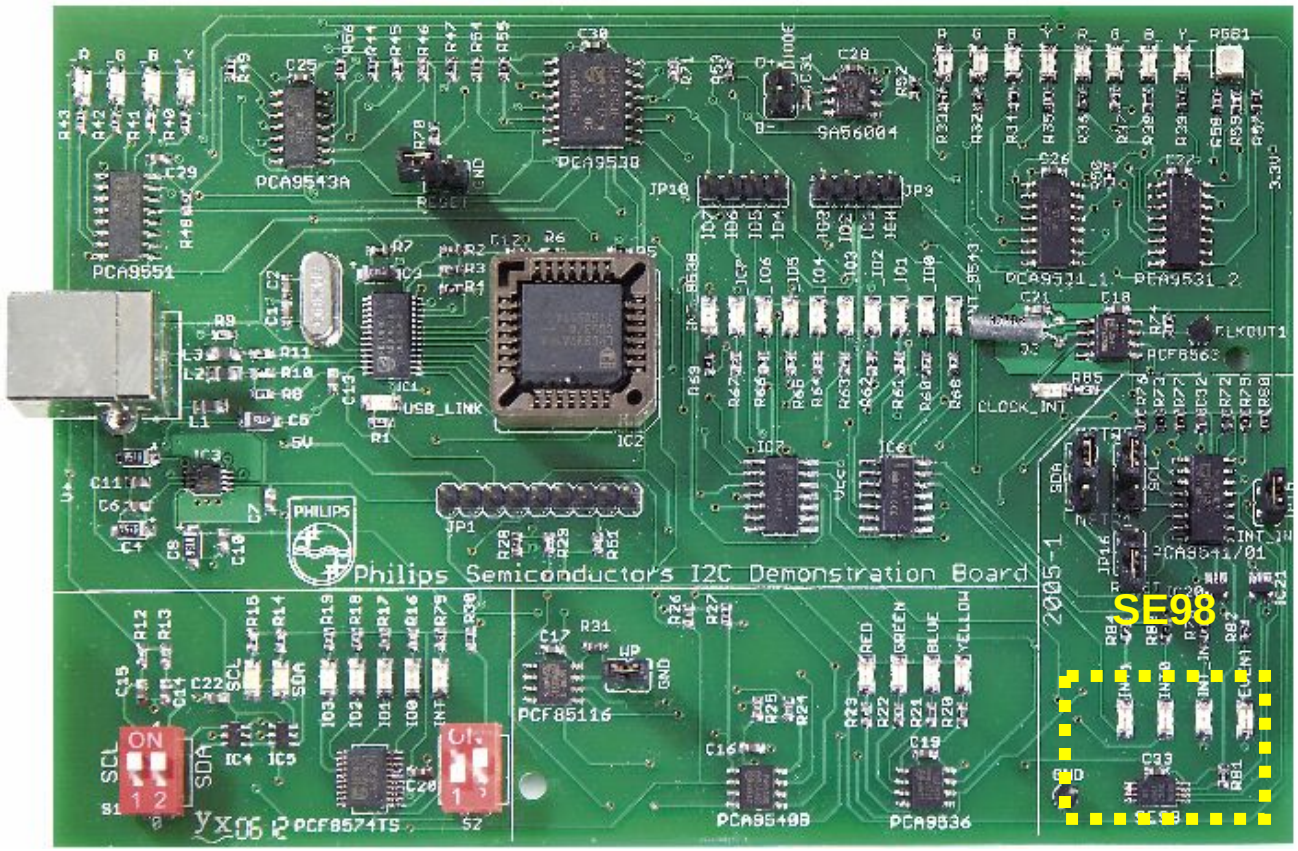
# I²C - Temperature and Voltage Sensors

## Demoboard I2C 2005-1: SE98 schematic



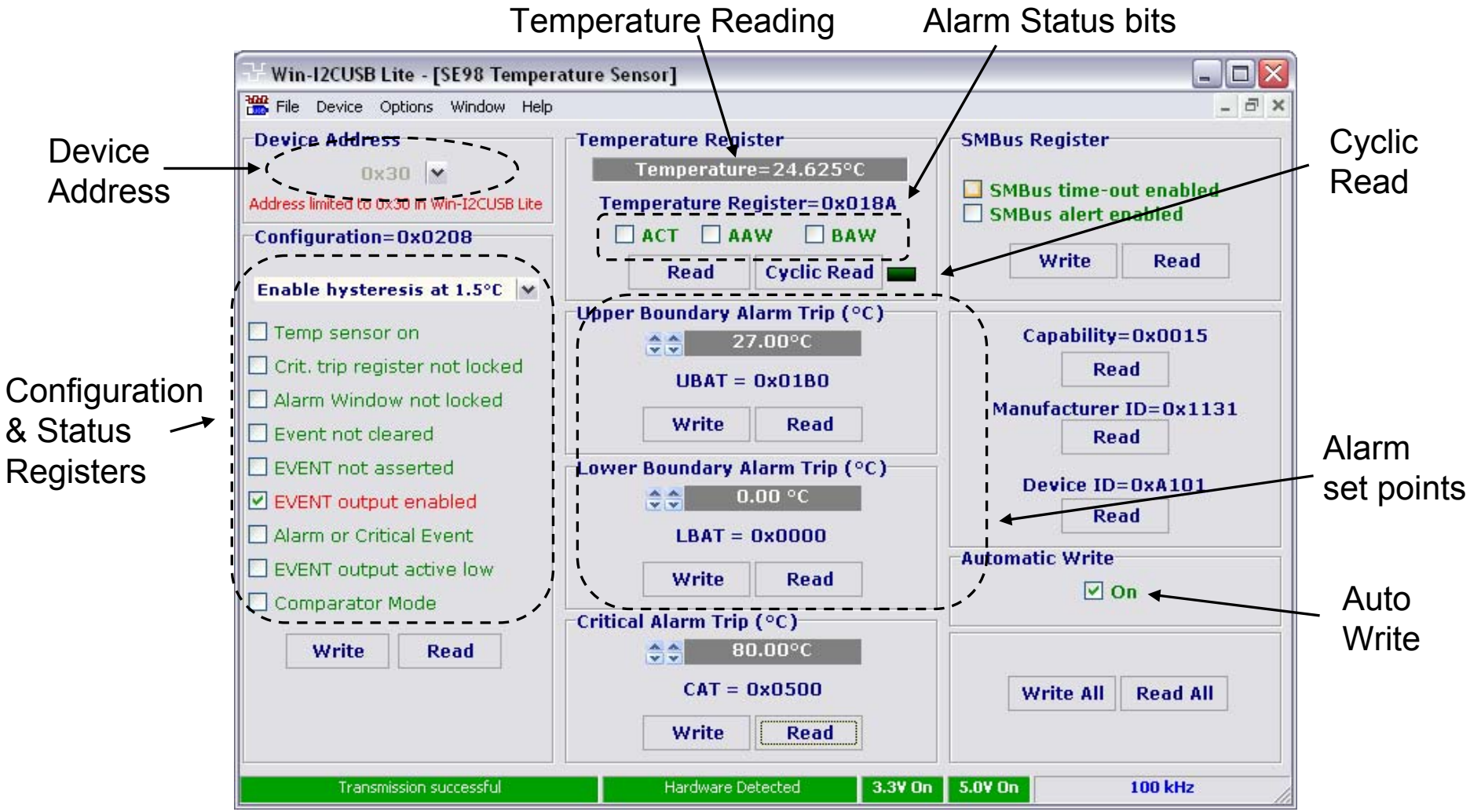
# I²C - Temperature and Voltage Sensors

## Demoboard 2005-1: SE98 on the board



# I²C - Temperature and Voltage Sensors

## Demoboard I2C 2005-1: SE98 GUI



# I²C - Temperature and Voltage Sensors

## Demoboard I2C 2005-1: Exercise A: SE98, Device Initialization

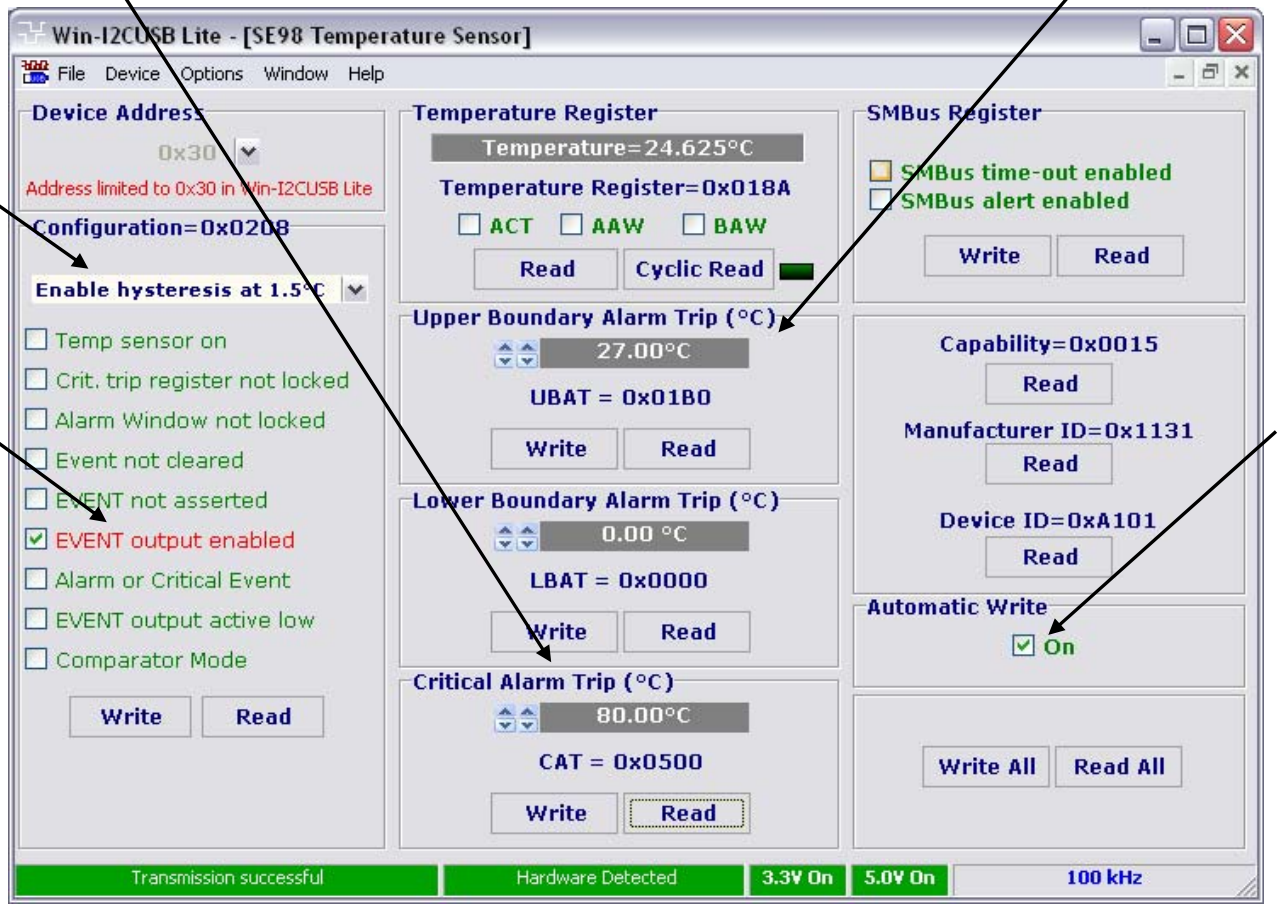
Set critical boundary alarm trip to 80 °C boundary registers

Set upper boundary alarm trip to 27 °C

Set hysteresis to 1.5 °C

Enable /EVENT output

Auto Write ON



# I²C - Temperature and Voltage Sensors

## Demoboard I2C 2005-1: Exercise A: SE98, EVENT output in Comparator vs Interrupt Mode

1. Visually check EVENT LED - it should be OFF.
2. Click on the “Cyclic read” for continuous temperature update.
3. Heat up or place your finger over the SE98 for a few minutes for temperature to exceed 28 °C - the EVENT LED is ON.
4. Cool down the SE98 or remove your finger from the SE98 and wait for a 1 minute or 2 until the temperature drops below 28 °C. What happens to the EVENT LED now? - EVENT LED is OFF.
5. Change the EVENT output to Interrupt mode and repeat steps 1-4. What happens to the EVENT LED now? - EVENT LED is still ON.
6. Clear the EVENT bit. - EVENT LED is OFF.

Conclusion:

- Comparator mode, the EVENT output de-asserted and LED is OFF when temperature falls within the ALARM window.
- Interrupt mode, the EVENT output stays asserted and LED is ON even after the temperature drops back within the ALARM window. The output is cleared by writing to the “event clear bit”.

# I²C - Temperature and Voltage Sensors

## Demoboard I2C 2005-1: Exercise A (cont.): SE98

1. Read the upper, lower and critical alarm boundary trips registers. What are their values?
2. Upper = 28 °C; Lower = 0 °C; Critical = 85 °C
3. Change the upper, lower and critical alarm boundary trip to 35 °C, -20 °C, and 125 °C and read them back. What are their values?
   Upper = 35 °C
   Lower = - 20 °C
   Critical = 125 °C
4. Check the “lock the Critical” and “lock alarm window” bits
5. Change the upper, lower, and critical alarm boundary trips to 28 °C, 0 °C, and 85 °C, and read them back. What are their values? No change
6. Remove the power will reset remove the lock feature

Conclusion: lock bit protects alarm set point values. Once lock bit is written, you can only clear it by powering off the SE98.

# I²C - Temperature and Voltage Sensors

## Demoboard 2005-1: Exercise B: SE98, Expert mode

- Power off and on the board
- Use the Expert Mode, write and verify the following codes:
  1. Set the upper alarm to 27 °C
  2. Confirm what you wrote is correct
  3. Set the critical temperature to 70 °C
  4. Confirm what you wrote is correct
  5. Write to configuration register to enable the EVENT output
  6. Read what you've just wrote
  7. Heat up SE98 and read the configuration register
  8. Read the temperature value

Hint: Next three slides show register definition

# I²C - Temperature and Voltage Sensors

## Demoboard 2005-1: Exercise B (cont.): SE98, Expert mode

Above Critical indicates whether the temperature exceeds the critical temperature
0: temp is below critical temperature
1: temp exceeds critical temperature

Sign bit
0: positive temperature
1: negative temperature

Above ALARM window indicates whether the temperature exceeds the alarm temperature
0: temperature is below the upper alarm
1: temperature exceeds the upper alarm

Below ALARM Window indicates whether the temperature falls below the lower alarm
0: temperature is above the lower alarm
1: temperature falls below the Lower ALARM

Table 15. Temperature Register bit allocation

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Symbol | ACT | AAW | BAW | SIGN | TEMP | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Access | R | R | R | R | R | R | R | R |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Symbol | TEMP | | | | | | | RFU |
| Reset | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| Access | R | R | R | R | R | R | R | R |

Integer

Decimal

Reserved for future use

Temperature Value = 25.125 °C

# I²C - Temperature and Voltage Sensors

## Demoboard I2C 2005-1: Exercise B (cont.): SE98, Register summary

**Table 3. Register summary**

| Address (hex) | POR state (hex) | Register name |
|---|---|---|
| n/a | n/a | Pointer Register |
| 00h | 0015h/0017h | Capability Register (B-grade = 0017h, C-grade = 0015h) |
| 01h | 0000h | Configuration Register |
| 02h | 0000h | Upper Boundary Alarm Trip Register |
| 03h | 0000h | Lower Boundary Alarm Trip Register |
| 04h | 0000h | Critical Alarm Trip Register |
| 05h | n/a | Temperature Register |
| 06h | 1131h | Manufacturer ID Register |
| 07h | A101h | Device ID/Revision Register |
| 08h to 21h | 0000h | reserved registers |
| 22h | 0000h | SMBus Register |
| 23h to FFh | 0000h | reserved registers |

# I²C - Temperature and Voltage Sensors

## Demoboard 2005-1: Exercise B (cont.): SE98, Configuration register

Critical Lock Bit
0: alarm register not locked, register value can be changed
1: alarm register locked , register value cannot be changed

ALARM Lock Bit
0: Alarm not locked, upper and lower alarm can be modified
1: Alarm locked, upper and lower alarm cannot be modified

Clear EVENT flag
0: no effect
1: clear the event flag in interrupt mode

Hysteresis for alarm
00: hysteresis disabled
01: hysteresis = 1.5 °C
10: hysteresis = 3 °C
11: hysteresis = 6 °C

0: Temp sensor shutdown mode
1: Normal mode

Table 6. Configuration Register (address 01h) bit allocation

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Symbol | | | RFU | | | HEN | | SHMD |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Access | R | R | R | R | R | R/W | R/W | R/W |
| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Symbol | CTLB | AWLB | CEVNT | ESTAT | EOCTL | CVO | EP | EMD |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

EVENT Status bit
0: not asserted
1: asserted

EVENT output control bit
0: EVENT output disabled
1: EVENT output enabled

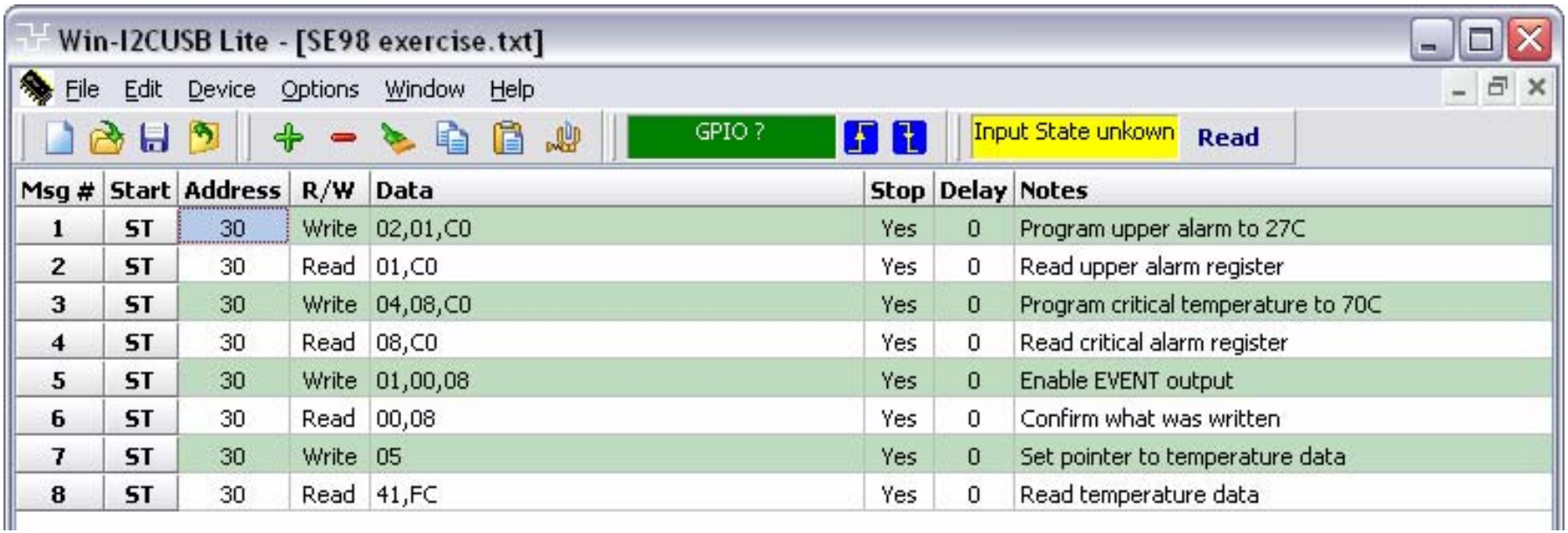
Critical EVENT Only
0: Critical+Alarm
1: Critical

EVENT Polarity
0: Active low
1: Active high

EVENT Mode
0: Comparator mode
1: Interrupt mode

# I²C - Temperature and Voltage Sensors

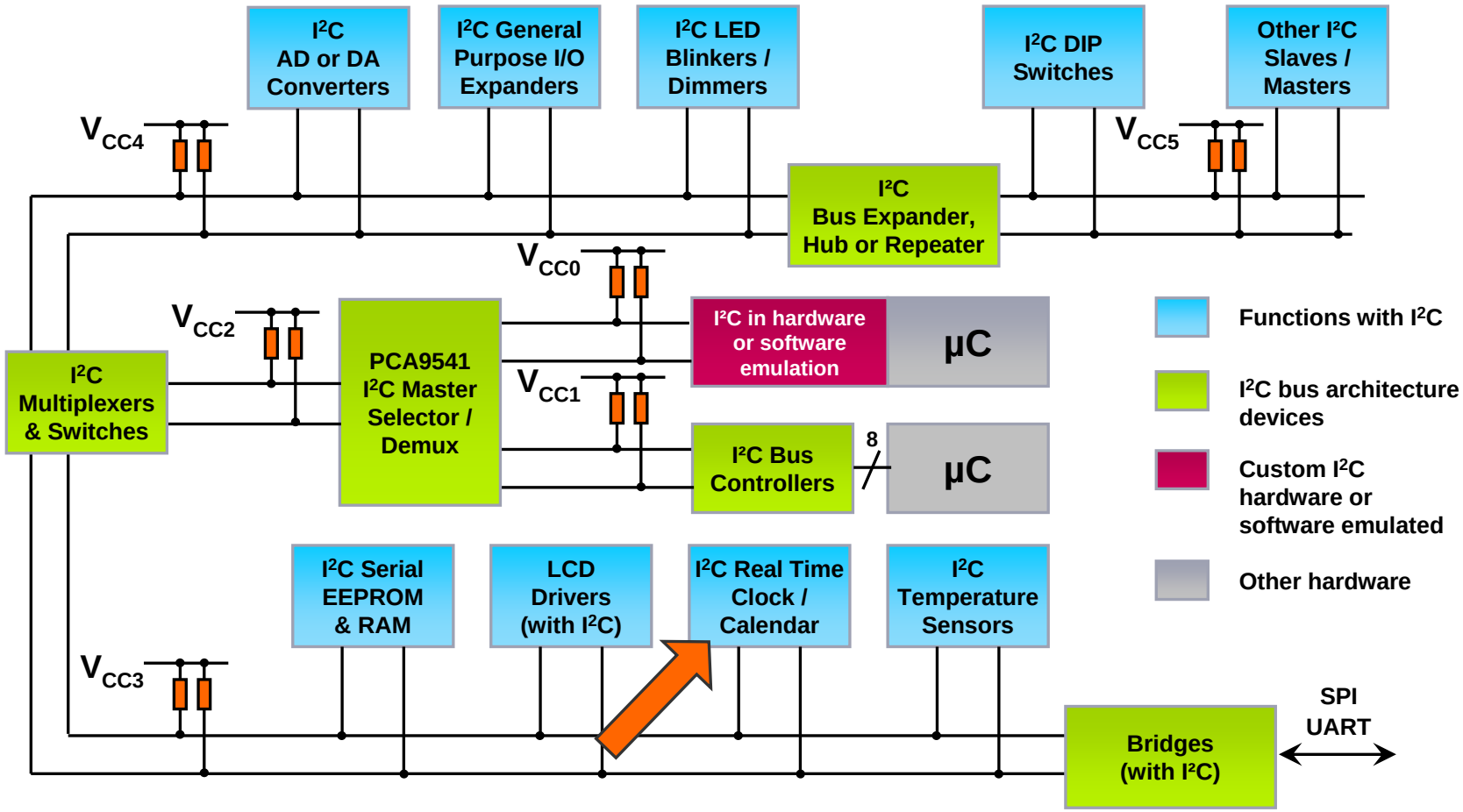
## Demoboard I2C 2005-1: Exercise B (cont.): SE98, solution




| Msg # | Start | Address | R/W | Data | Stop | Delay | Notes |
|---|---|---|---|---|---|---|---|
| 1 | ST | 30 | Write | 02,01,C0 | Yes | 0 | Program upper alarm to 27C |
| 2 | ST | 30 | Read | 01,C0 | Yes | 0 | Read upper alarm register |
| 3 | ST | 30 | Write | 04,08,C0 | Yes | 0 | Program critical temperature to 70C |
| 4 | ST | 30 | Read | 08,C0 | Yes | 0 | Read critical alarm register |
| 5 | ST | 30 | Write | 01,00,08 | Yes | 0 | Enable EVENT output |
| 6 | ST | 30 | Read | 00,08 | Yes | 0 | Confirm what was written |
| 7 | ST | 30 | Write | 05 | Yes | 0 | Set pointer to temperature data |
| 8 | ST | 30 | Read | 41,FC | Yes | 0 | Read temperature data |

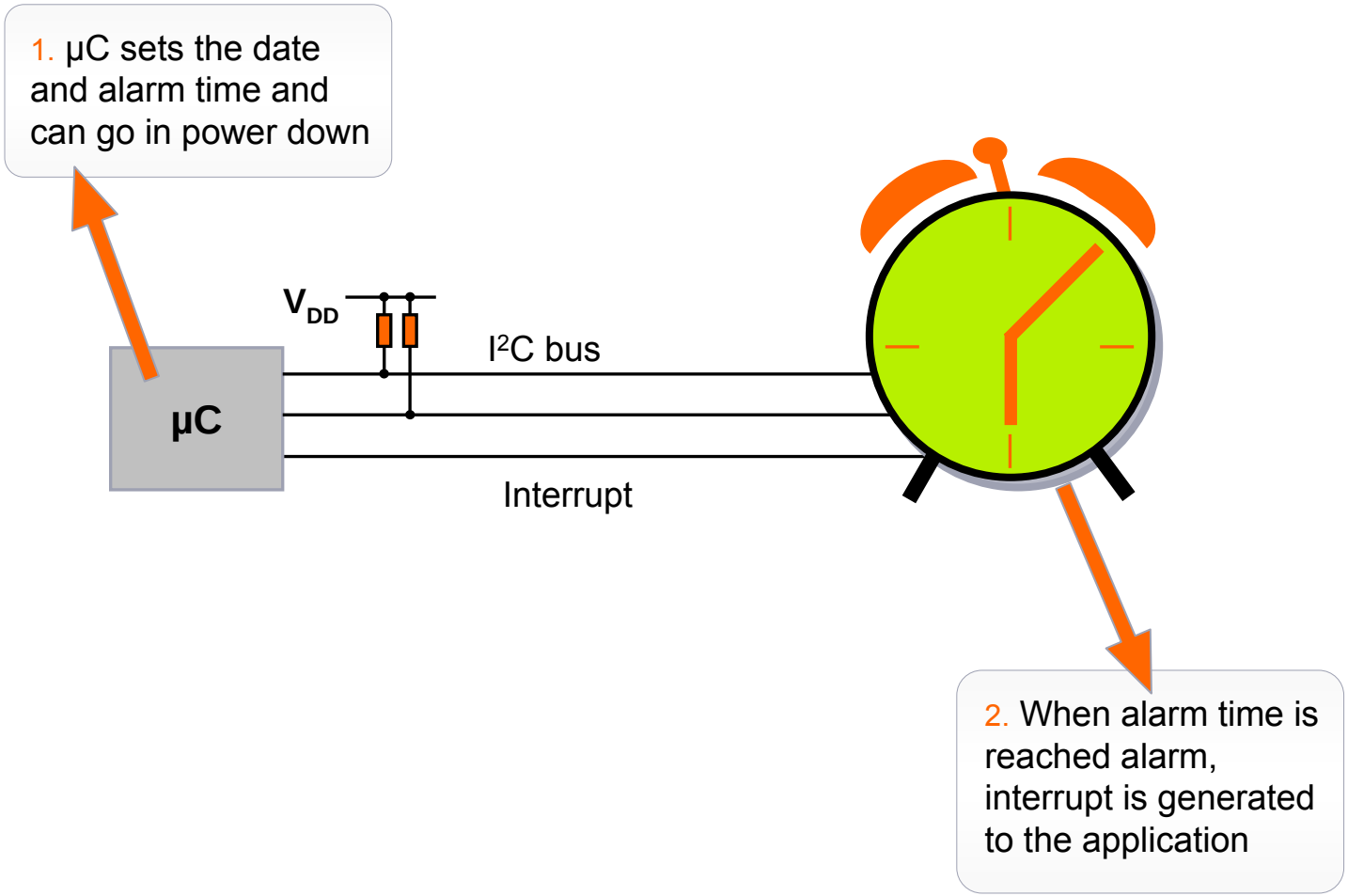
# I²C - Real Time Clocks / Calendars

# I²C - Real Time Clocks / Calendars



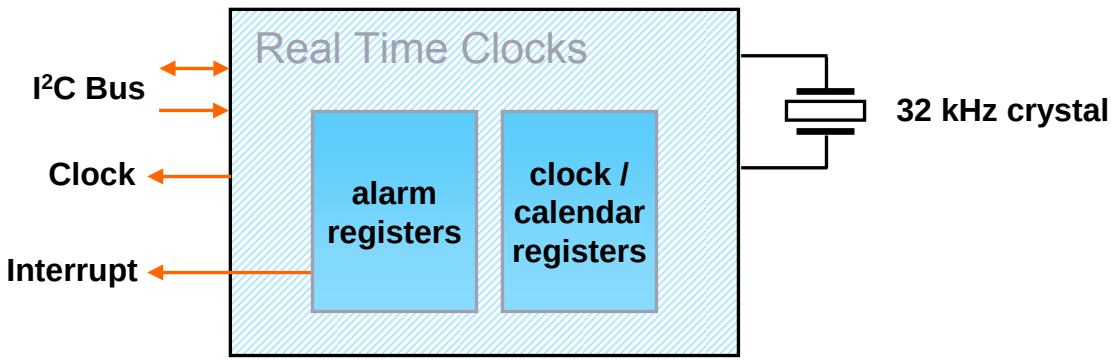
# I²C - Real Time Clocks / Calendars

## Application



# I²C - Real Time Clocks / Calendars

## Product overview



# I²C - Real Time Clocks / Calendars

## PCA8565 / PCF8563 / PCF8583 / PCF8593


General:

- Time keeping
- Large voltage range 1.0 .. 5.5 V
- Clock from seconds to 99 years
- Programmable Timer
- Frequency output
- Small packages (TSSOP8, HVSON10, etc)

Application:

- Time keeping, calendar
- Tariff switching
- Integrity check after power-down

Products:

- PCA8565:
  - High temperature (up to 125 °C)
- PCF8563:
  - Low power: < 250 nA
- PCF8583:
  - 240-byte RAM
- PCF8593:
  - 1/100-second resolution

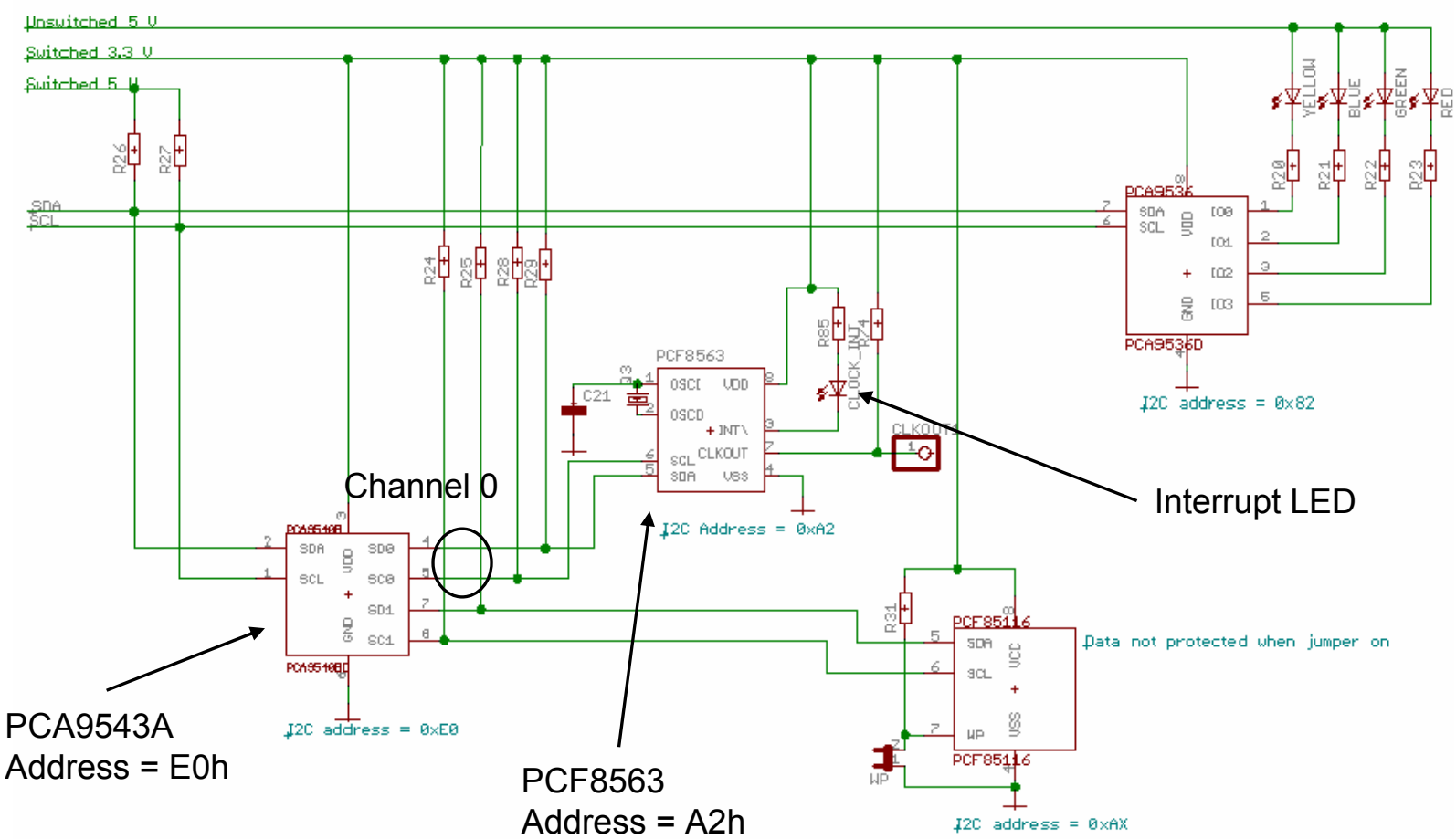
# I²C - Real Time Clocks / Calendars

## Product summary

| Features | PCA8565 | PCF8563 | PCF8583 | PCF8593 |
|---|---|---|---|---|
| I2C-bus interface speed | 400 kHz | 400 kHz | 100 kHz | 100 kHz |
| Scratch pad RAM | 0 | 0 | 248 Bytes | 0 |
| Years and leap year tracking | yes | yes | yes | yes |
| Year counter | 2 digit +1 bit | 2 digit +1 bit | 2 bit | 2 bit |
| 1/10, 1/100 s counter | no | no | 10 ms | 10 ms |
| Programmable alarm | yes | yes | yes | yes |
| Low voltage detector | yes | yes | no | no |
| Supply voltage for I2C-bus | 1.8 - 5.5 V | 1.8 - 5.5 V | 2.5 - 6.0 V | 2.5 - 6.0 V |
| Supply voltage for clock | 1.8 - 5.5 V | 1.0 - 5.5 V | 1.1 - 6.0 V | 1.1 - 6.0 V |
| Typical power consumption | 700 nA @ 2 V | 250 nA @ 1 V | 2 µA @ 1.0 V<br>10 µA@ 5 V | 1 µA @ 2.0 V<br>4 µA @ 5 V |
| Packages | -40 - 125°C | -40 - 85°C | -40 - 85°C | -40 - 85°C |

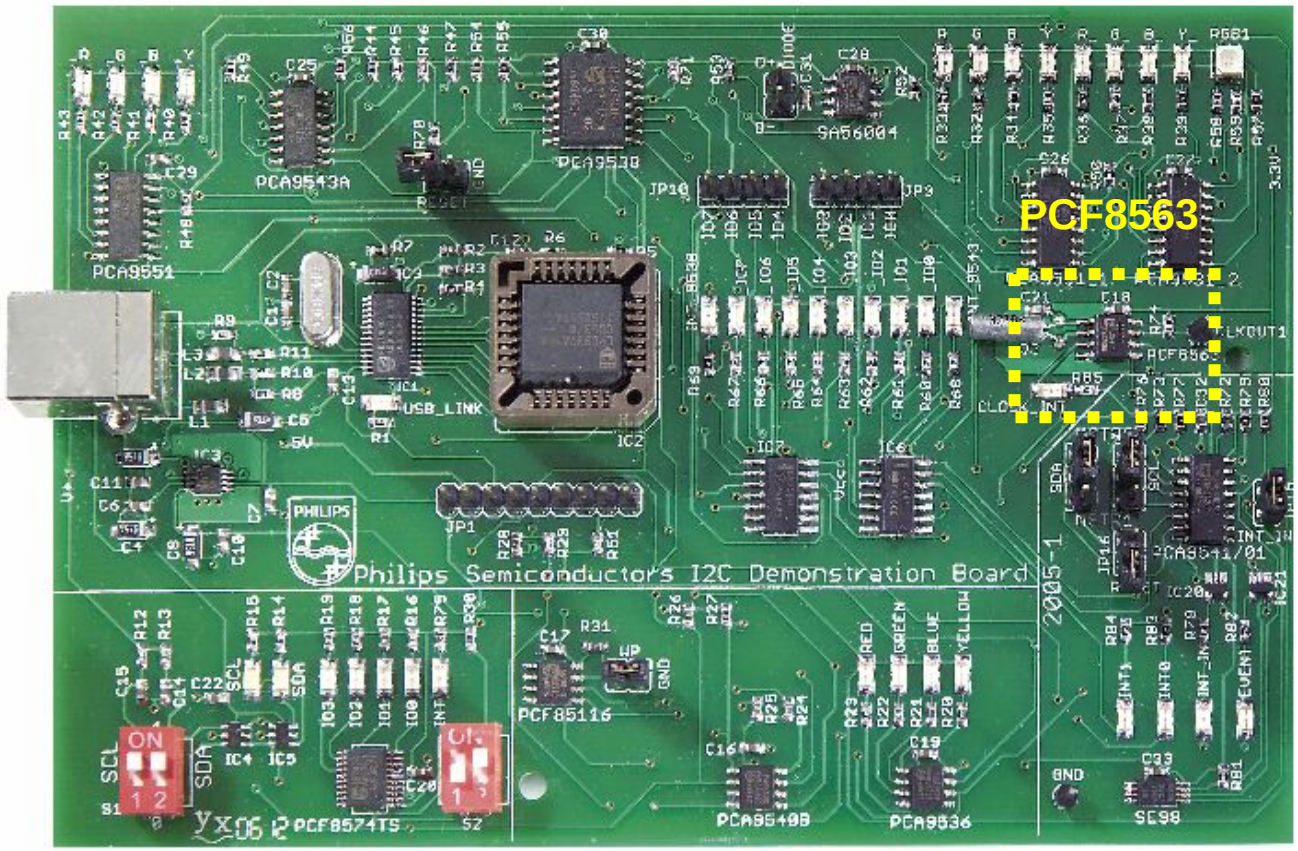
# I²C - Real Time Clocks / Calendars

## Demoboard I2C 2005-1: PCF8563 Schematic



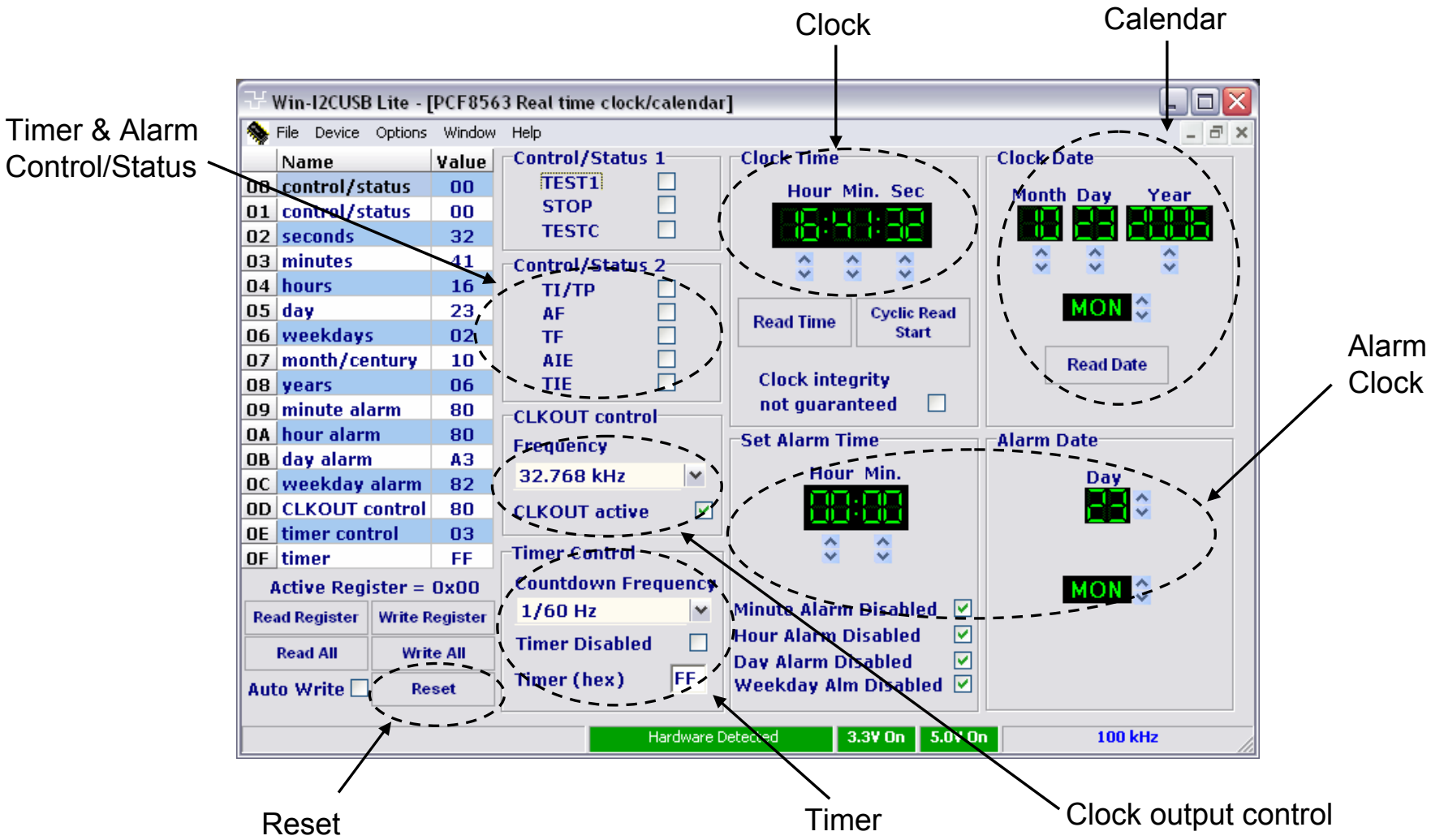
# I²C - Real Time Clocks / Calendars

## Demoboard I2C 2005-1: PCF8563 on the board



# I²C - Real Time Clocks / Calendars

## Demoboard I2C 2005-1: PCF8563 GUI



# I²C - Real Time Clocks / Calendars

## Demoboard I2C 2005-1: Exercise A: PCF8563 Alarm Mode

1. Click on the reset push button to reset alarm, timer, clock and calendar
2. Alarm mode: Minute, hour, day, or week
3. Click on the cyclic read push button
4. Select the alarm minute enable checkbox
5. Select the alarm interrupt enable (AIE) checkbox
6. Set alarm to 1 minute
7. Do a “write all” on the screen to update the register
8. What do you see when the clock time reaches 1 minute?
9. Interrupt is asserted and interrupt LED is on
10. Alarm flag (AF) status bit is set
11. Clear AF (by uncheck the box and do a “Write All”)

# I²C - Real Time Clocks / Calendars

## Demoboard I2C 2005-1: Exercise B: PCF8563 Timer Mode

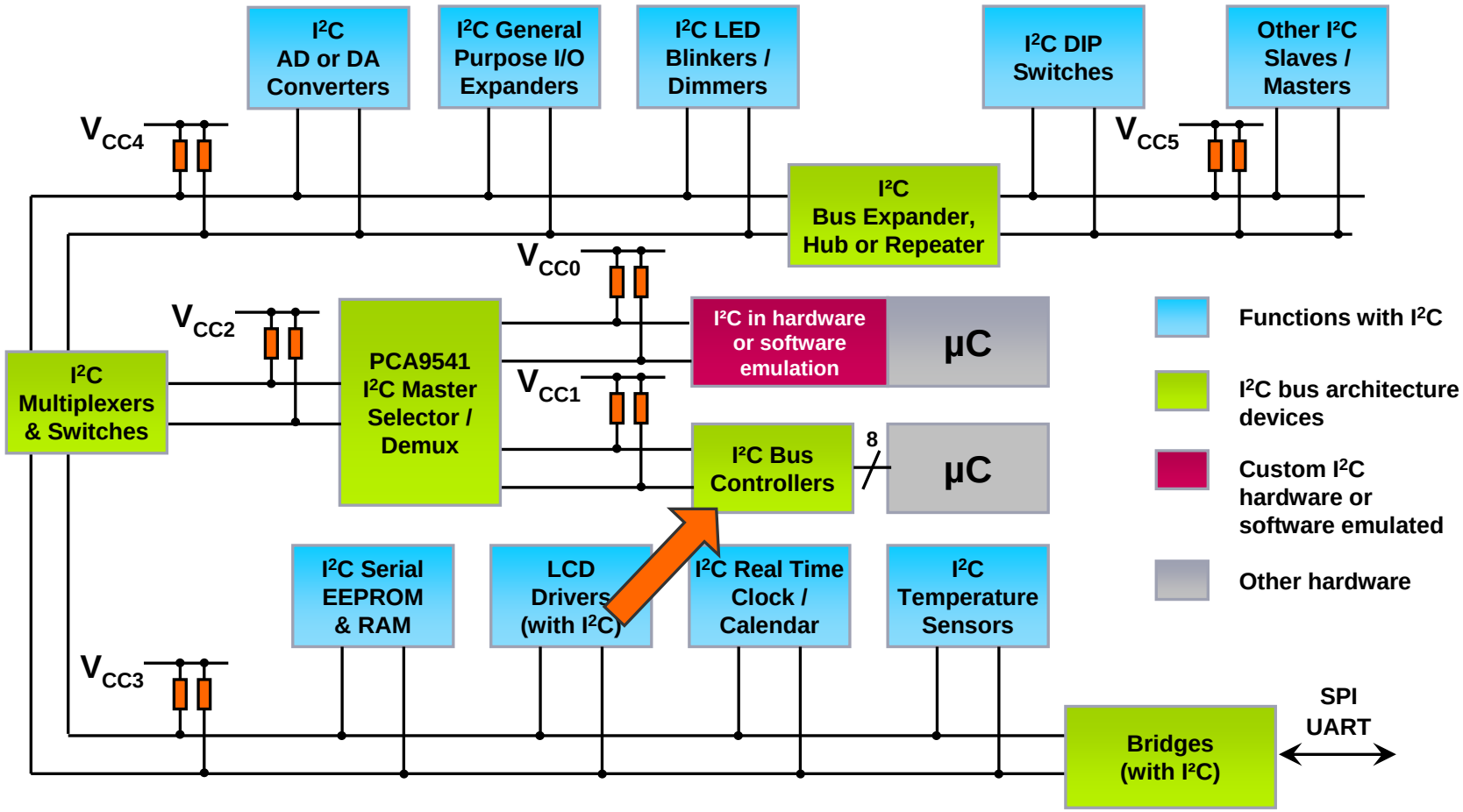
1. The timer has a selectable count down frequency ranging from 1/64 Hz to 4096 Hz, and an 8-bit or 256 count down value
2. Chose 1 Hz (1 sec) timer count down frequency
3. Program a count down value of 12
4. Select timer interrupt enable (TIE) checkbox
5. Enable timer
6. Perform a “Write All“ command
7. Wait for 12 seconds, read the status register, and observe the Interrupt LED light
8. What happens to TF and LED?
9. TF is set
10. LED is ON

# $I^2C$ - Portfolio: bus architecture devices

# I²C - Bus Enablers

- I²C bus controllers
- I²C master selector / demux
- I²C multiplexers & switches
- I²C bus expander, hub or repeater
- Bridges (with I²C)

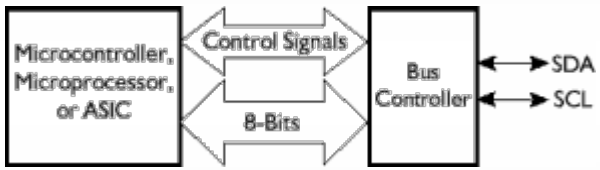
# I²C - Bus controllers



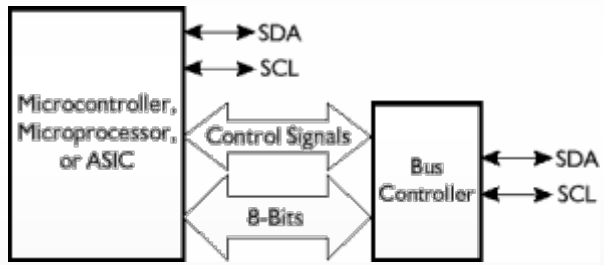
# I²C - Bus controllers

## Application

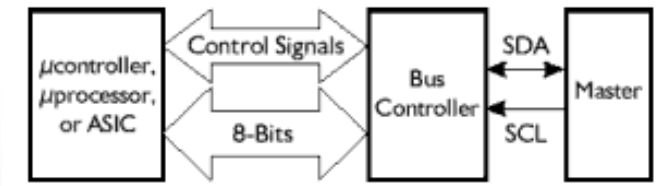
### Adding an I²C bus port




### Adding additional I²C bus ports

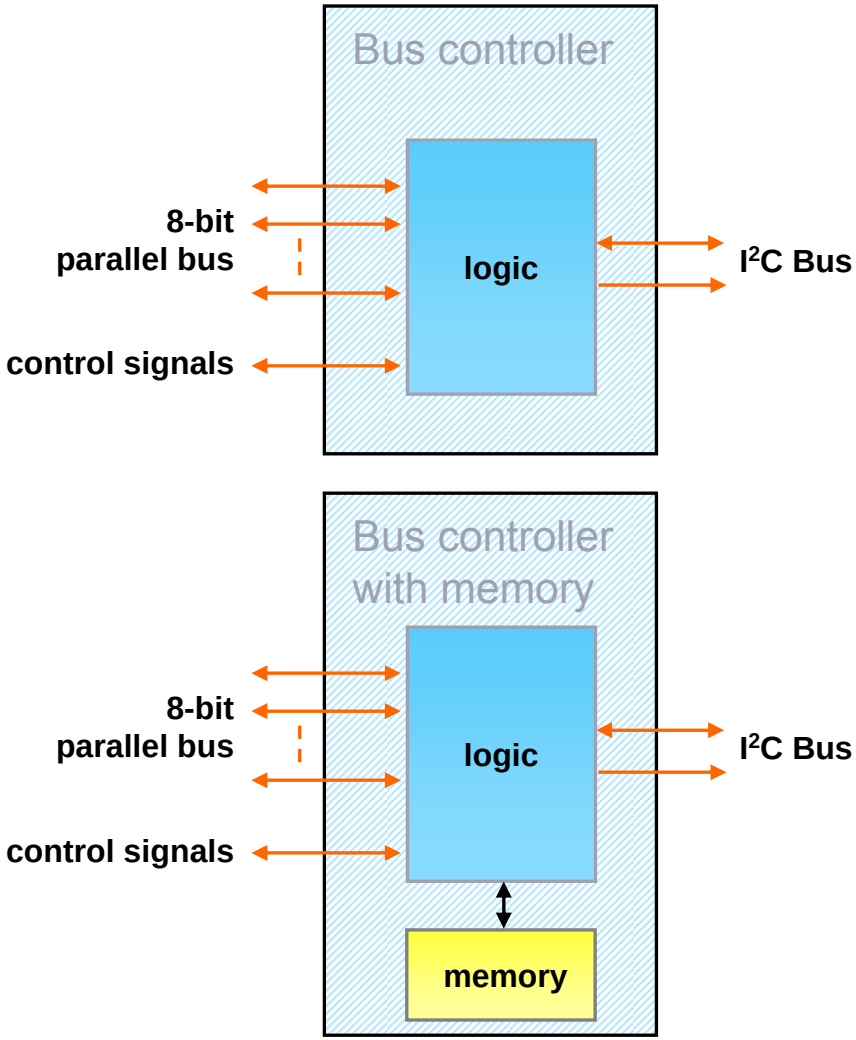



### Converting 8 Bits of Parallel Data into I2C Serial Data Stream



# I²C - Bus controllers

## Product overview

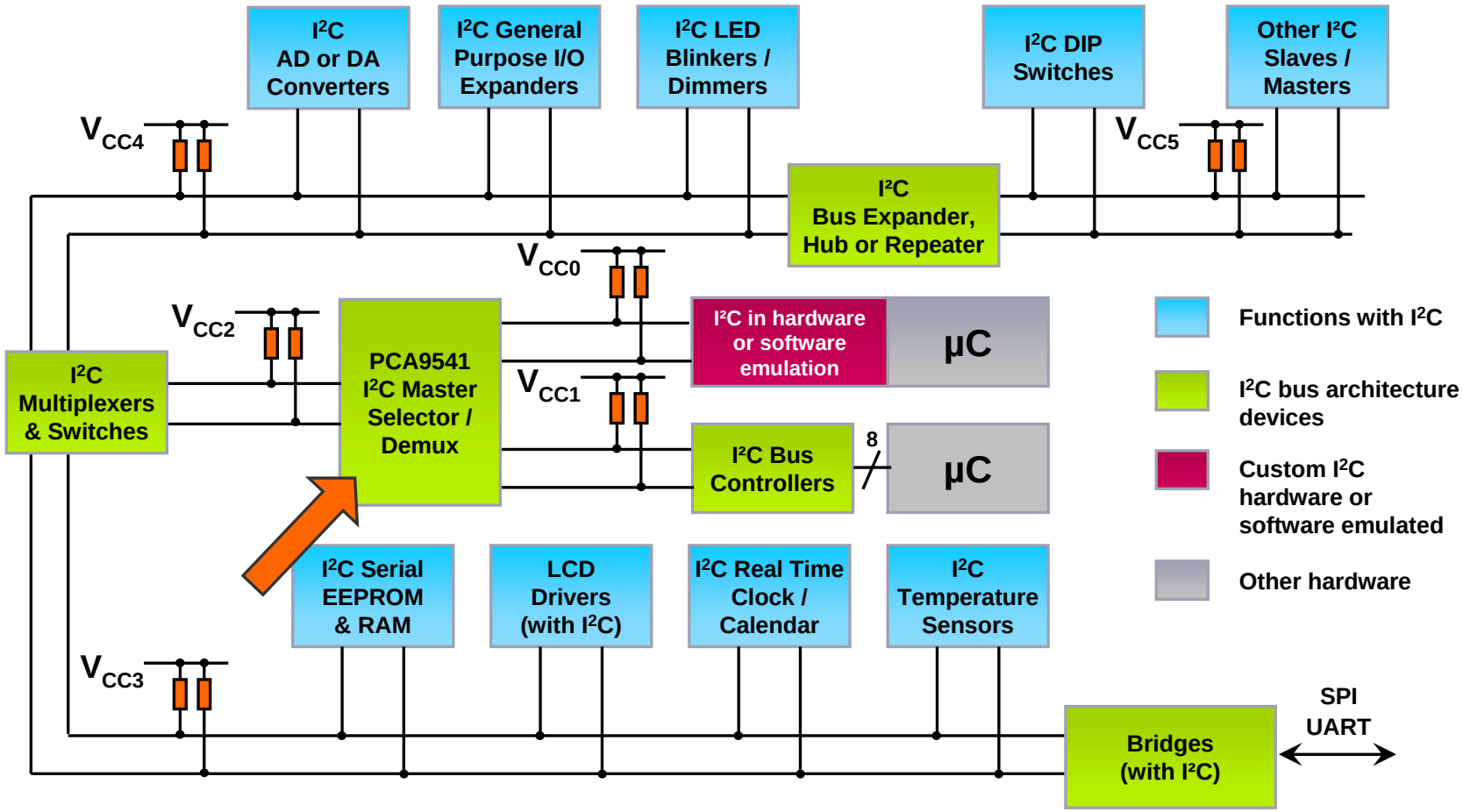


# I²C - Bus controllers

## Product summary

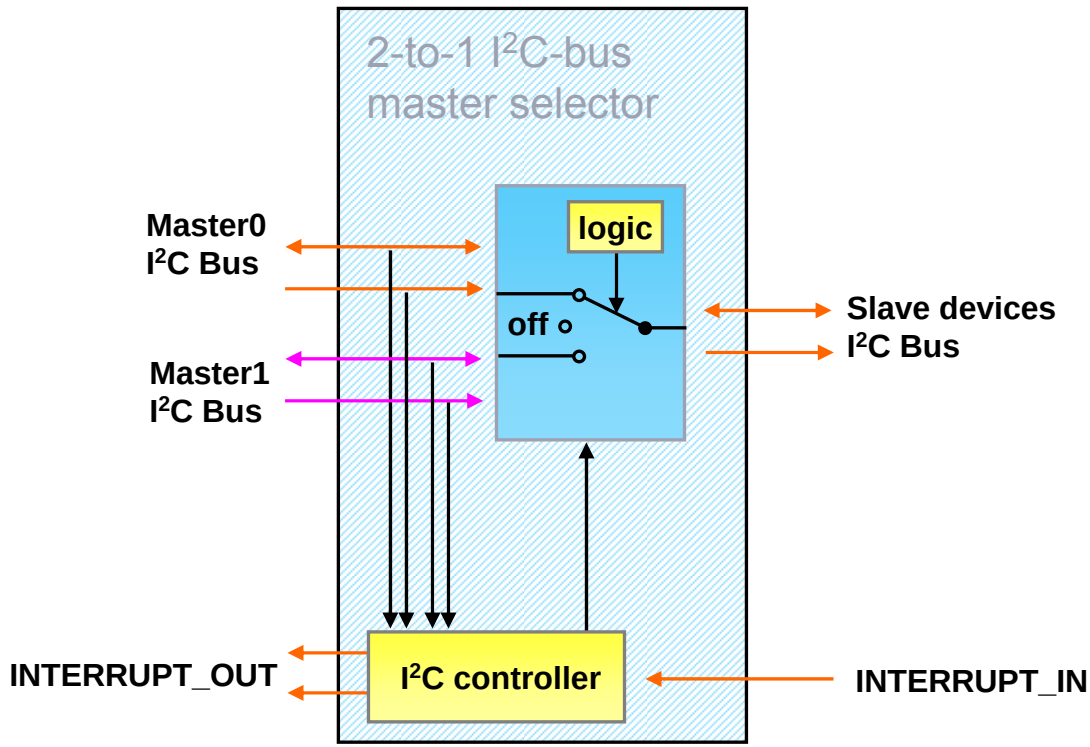
| | Voltage Range | Max I²C Freq. | Clock Source | Parallel Interface |
|---|---|---|---|---|
| PCF8584 | 4.5V - 5.5V | 90kHz | External | Slow |
| PCA9564 | 2.3V - 3.6V; with 5V Tolerance | 360kHz | Internal | Fast |
| PCA9665 | 2.3V - 3.6V; with 5V Tolerance | 1000kHz | Internal (Trimmed) | Fast with 68 Byte Buffer |

# I²C - Master selector / demux



# I²C - Master selector / demux

## Product overview



# I²C - Master selector / demux

## PCA9541 - 2-to-1 I²C-bus master selector

### Features

- Channel selection via I²C -bus
- Downstream bus initialization/recovery function before changing channel
- Bus traffic sensor
- No glitch on power-up and supports hot insertion
- Software identical for both masters
- Voltage translation between 1.8 V, 2.5 V, 3.3 V and 5 V buses
- Operating power supply voltage range of 2.3 V to 5.5 V
- 4 address pins allowing up to 16 devices on the I²C-bus
- 0 to 400 kHz I²C clock frequency

### Applications

- High-End Servers, Base Stations and Mass Storage systems with high maintenance requirements
- Two independent masters to a single I²C slave device

### Versions

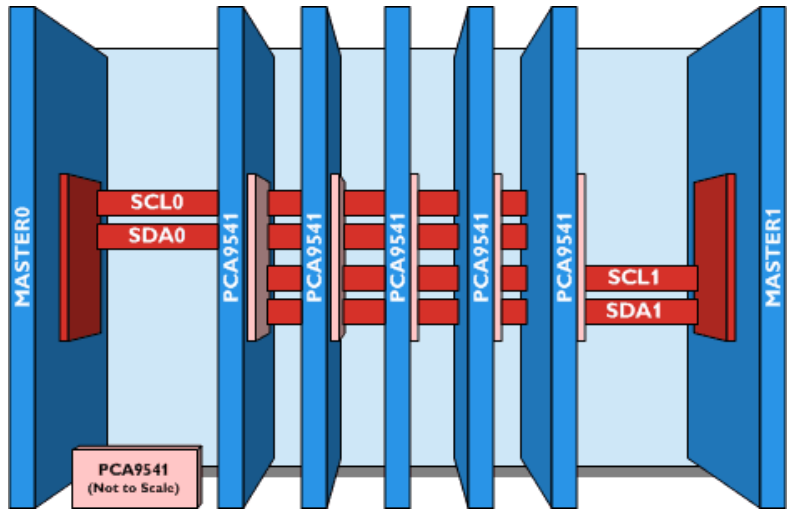
- PCA9541/01 - Master Channel 0 selected after power-up/reset
- PCA9541/03 - No master channel (off) selected after power-up/reset and either master can take control of the bus

# I²C - Master selector / demux

## PCA9541 - application

High Reliability Backplane Use

- In a typical multi-point application:
  - Two masters (e.g. primary and back-up) are located on separate I²C buses that connect to multiple downstream I²C bus slave cards/devices.
  - PCA9541/01: for non-hot swap applications  PCA9541/03: for hot swap applications
- If a master fails or the master card is removed for repair or service, the back-up master can take over
- Either master can take command of the I²C bus and, at any time, can gain control of the slave devices if the other master is unable to communicate.
- The failed master is isolated from the system and will not affect communication between the active master and the slave devices located on the cards.

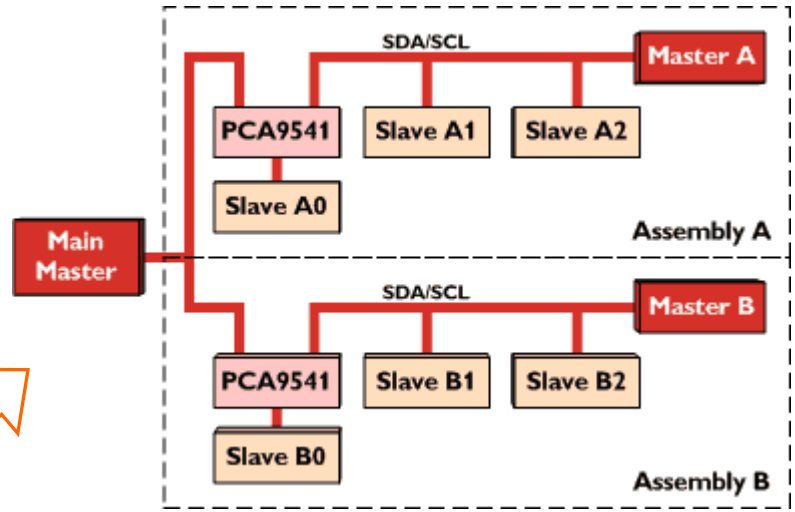


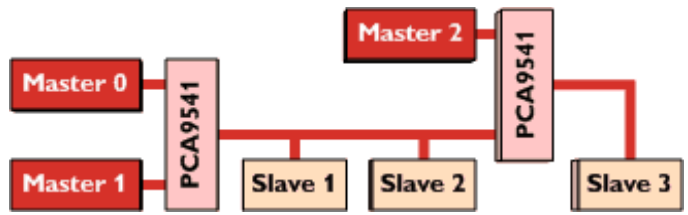
# I²C - Master selector / demux

## PCA9541 – application (2)

### Masters with Shared Resources Use

- Some masters may not be multi-master capable or some masters may not work well together and continually lock up the bus. The PCA9541 can be used to separate the masters but still allow shared access to slave devices such as Field Replaceable Unit (FRU) EEPROM or temperature sensors.
- If the /01 version is used, at power up:
  - Master A will control Slave A0 and Master B will control Slave B0
  - Main Master will be disconnected from both Assembly A and B
- If the /01 version is used, at power up:
  - Master 0 will control Slave 1 and Slave 2 (but not Slave 3)
  - Master 1 will be disconnected from the entire system Master 2 will control Slave 3



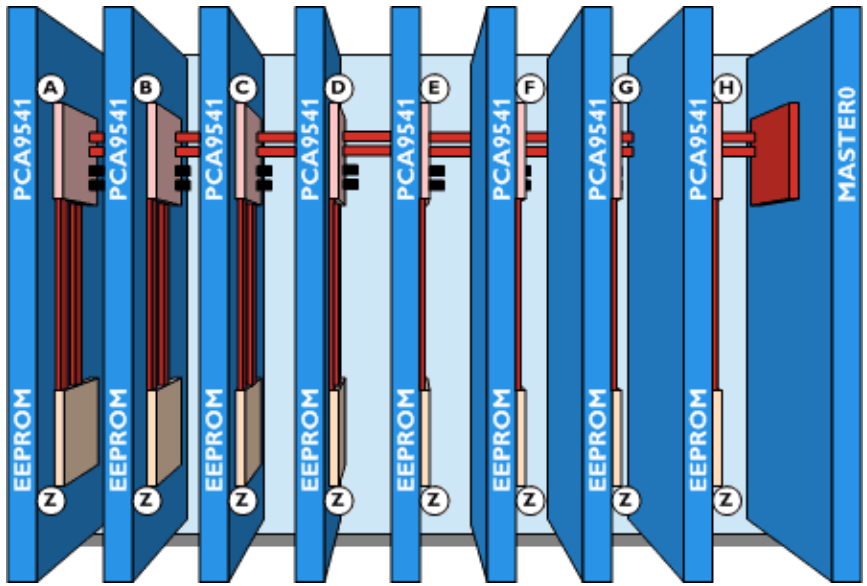



# I²C - Master selector / demux

## PCA9541 - application (3)

### Gatekeeper Multiplexer Use

- The PCA9541/03 can act as a gatekeeper multiplexer in applications where there are multiple I²C devices with the same fixed address (e.g., EEPROM with an address of "Z" as shown in the figure) connected in a multi-point arrangement to the same I2C bus.
- Up to 16 hot swappable cards/devices can be multiplexed to the same bus master by using one PCA9541/03 per card/device.
- Since each PCA9541/03 has its own unique address (e.g., "A", "B", "C", etc.), the EEPROM can be connected to the master, one at a time, by connecting one PCA9541/03 (Master 0 position) while keeping the rest of the cards/devices isolated (off position).
- Alternative: use PCA9548 1-to-8 channel switch on the master card and run 8 I²C buses

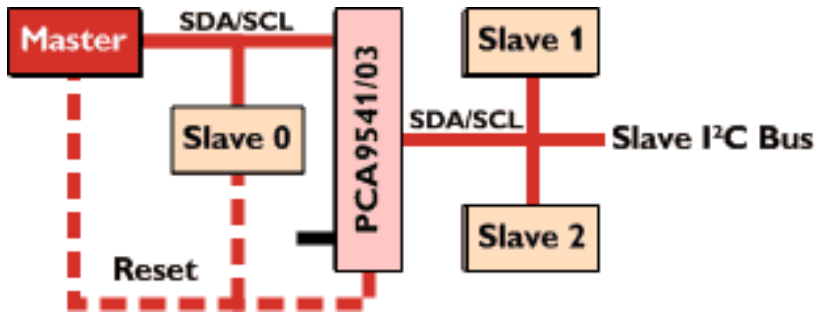


# I²C - Master selector / demux

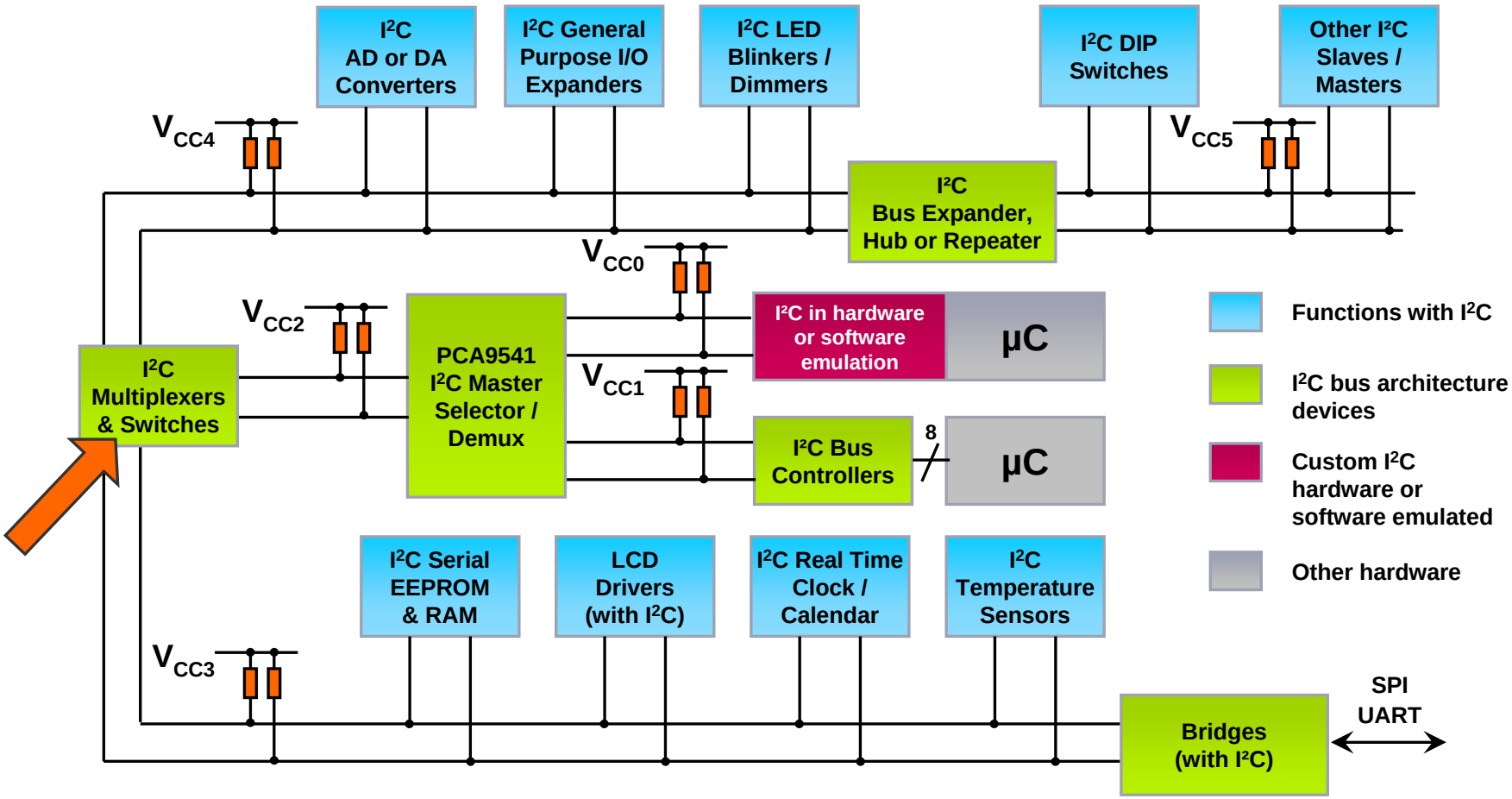
## PCA9541 - application (4)

Bus Initialization/Recovery Use

- If the I²C bus is hung, I²C devices without a hardware reset pin (e.g. Slave 1 and Slave 2 in the diagram) can be isolated from the master by the PCA9541/03.
- The PCA9541/03 disconnects the downstream bus when it is reset via the hardware reset line, restoring the master's control of the upstream bus (e.g. Slave 0). The bus master can then command the PCA9541/03 to send 9 clock pulses/stop condition to reset the downstream I²C devices before they are reconnected to the master.



# I²C - Multiplexers & switches



# I²C - Multiplexers & switches

## Application

### I²C Multiplexing

- Resolve address conflicts: multiplexers and switches split the I²C bus into several sub-branches and allow the I²C master to select and address one of multiple identical devices

### Capacitive Load Sharing

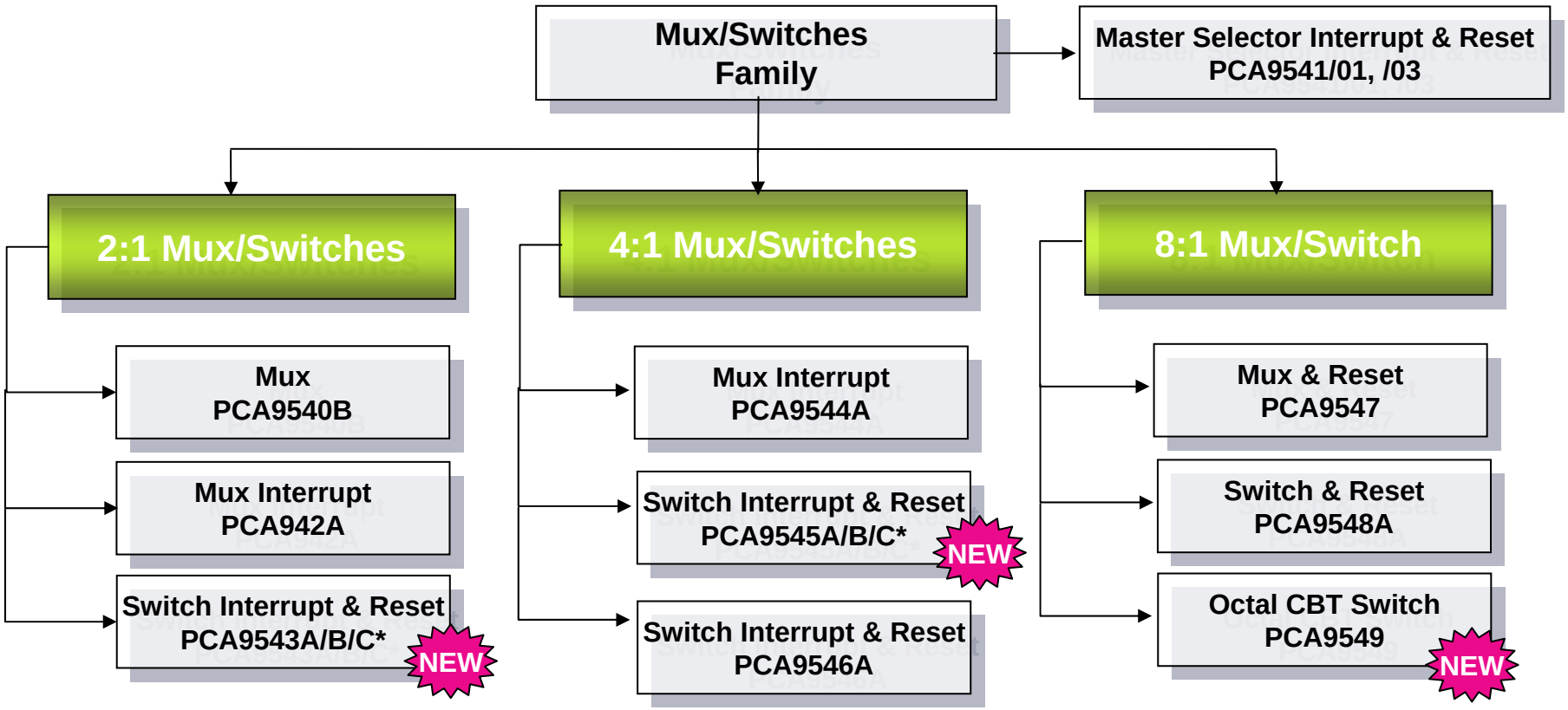
- Reduce capacitive bus loading by isolating the not needed bus segments
- When active, the channels act as a wire and the cumulative capacitive loading of the upstream channel and all active downstream channels must be considered.

### Voltage Level Shifting

- $V_{DD}$ pin can be used to limit the maximum high voltage passed by the device:
  - Allows different bus voltages on each pair
  - 2.3V or 3.3V devices can communicate with 5V devices without any additional protection. All I/O pins are 5V tolerant.
- Switches are best for voltage level shifting: multiple downstream channels can be active at the same time.

# I²C - Multiplexers & switches

## Product overview



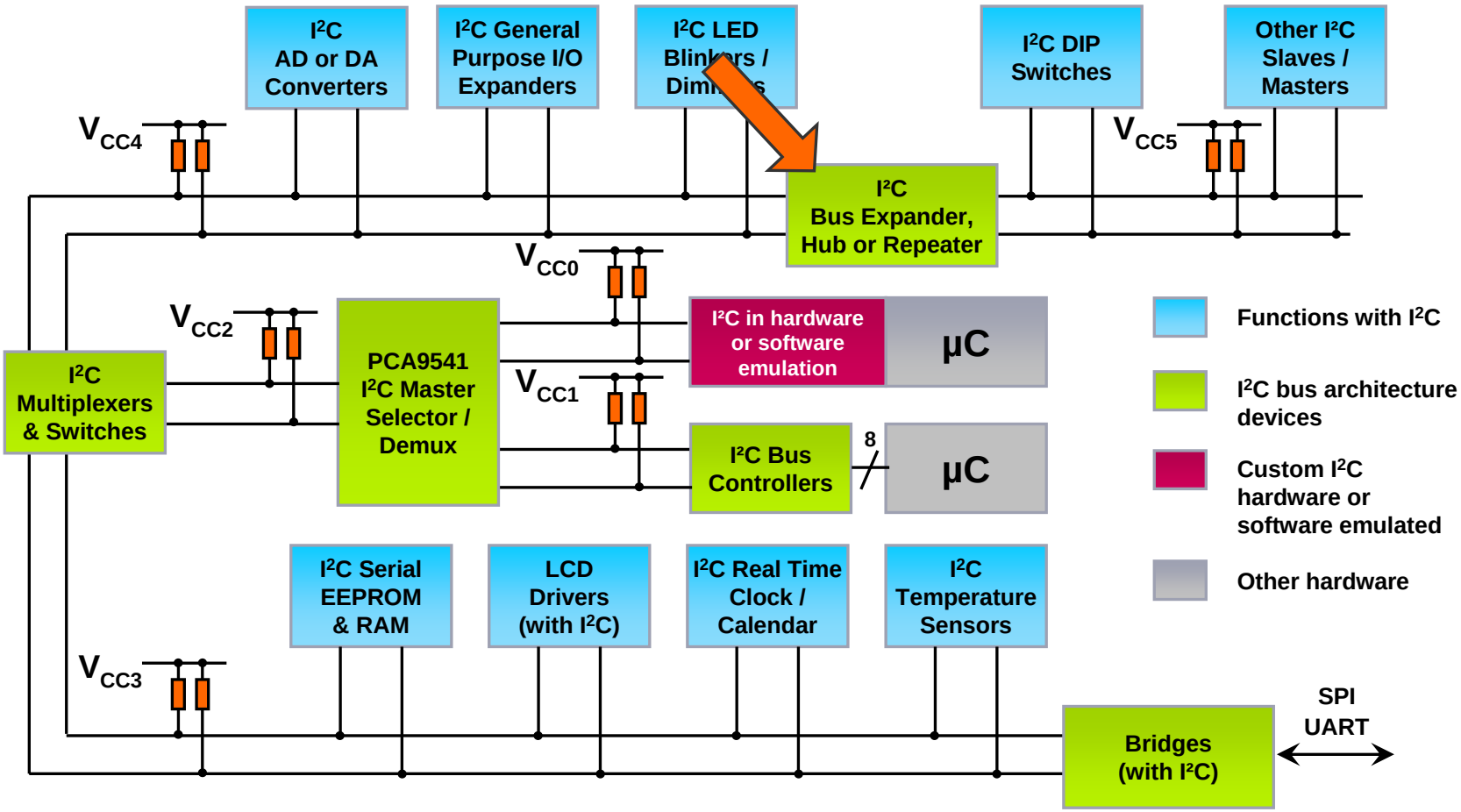

*Note: A, B, and C have different I²C fixed address

# I²C - Multiplexers & switches

## Product summary

| Device | Multiplexer (in/out) | Switch (in/out) | # of Addresses | Interrupt (in/out) | Hardware reset |
|---|---|---|---|---|---|
| PCA9540B | 1-2 | | 1 | | |
| PCA9541 | 2-1 | | 16 | 1-2 | ✓ |
| PCA9542A | 1-2 | | 8 | 2-1 | |
| PCA9543A | | 1-2 | 4 | 2-1 | ✓ |
| PCA9544A | 1-4 | | 8 | 4-1 | |
| PCA9545A | | 1-4 | 4 | 4-1 | ✓ |
| PCA9546A | | 1-4 | 8 | | ✓ |
| PCA9547 | 1-8 | | 8 | | ✓ |
| PCA9548A | | 1-8 | 8 | | ✓ |
| PCA9549 | | 1-8 | 8 | | ✓ |

# I²C - Bus expander, hub or repeater



# I²C - Bus expander, hub or repeater

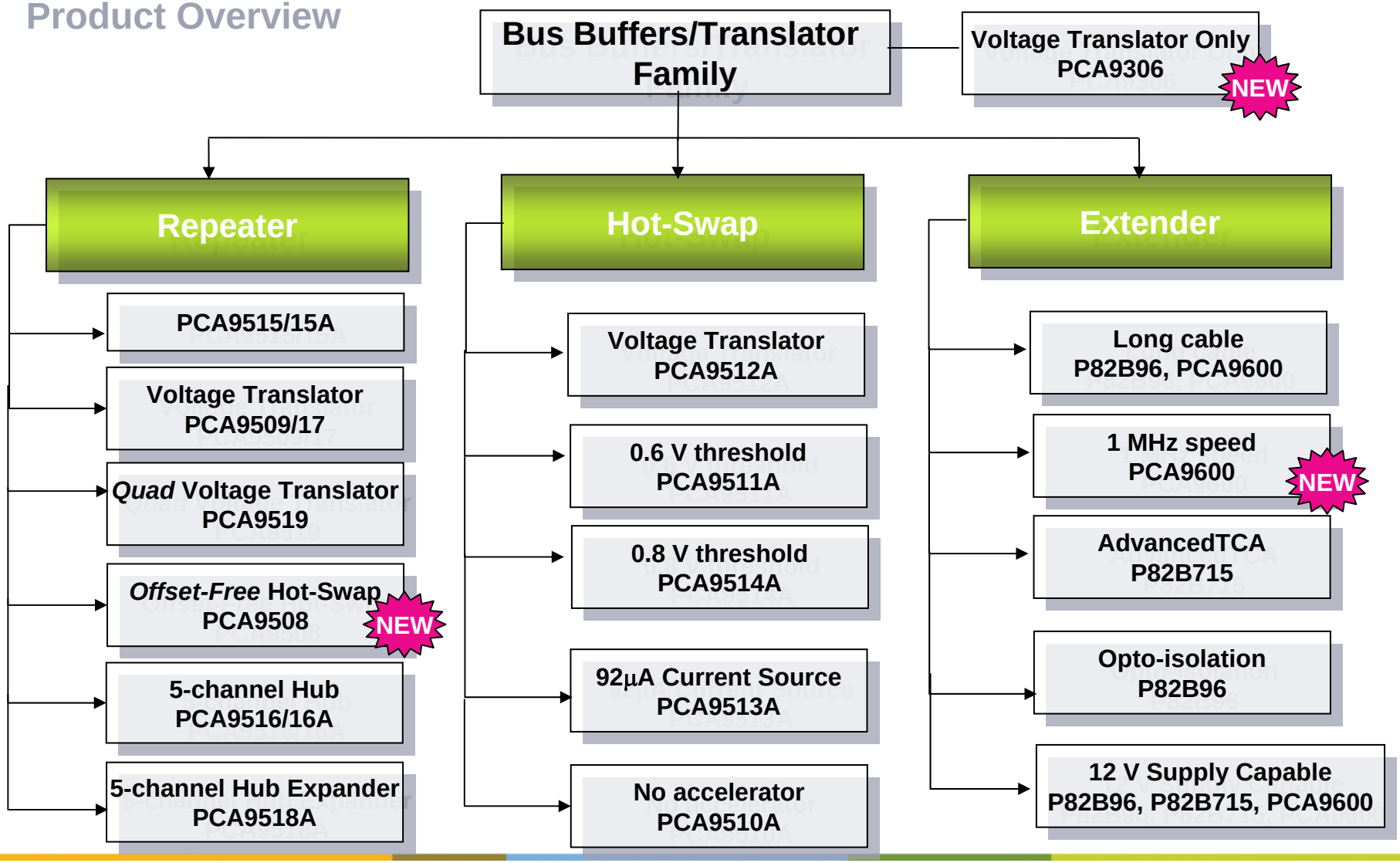
## When to use?

Bus expander, hub or repeater

- More I²C devices than the 400 pF maximum allowed by the I²C specification
- Long bus wiring in point-to-point or multi-point applications
- Different operating supply voltages or logic voltage levels within one system
- Opto-isolation for safety or due to difference in ground plane
- Isolating a section of a system that has lost its power supply
- Insertion of unpowered cards into an active I²C bus on multi-point backplanes like those found in CompactPCI, VME, or AdvancedTCA systems

# I²C - Bus expander, hub or repeater

## Product Overview



# I²C - Bus expander, hub or repeater

## PCA9507 versus PCA9517/17A

- PCA9507 is an upgrade replacement over PCA9517/17A with rise time accelerator for longer cable HDMI DDC application

PCA9507 and PCA9517/17A Comparison

| Features | PCA9507 | PCA9517A | PCA9517 |
|---|---|---|---|
| A-side with “rise time accelerator” for heavy load or long cable (up to 18 meters)* | Yes | No | No |
| A-side supply voltage range | 2.7V to 5.5V | 0.9V to 5.5V | 0.9V to 5.5V |
| ESD HBM | 5 kV | 5kV | 2kV |

* Note: Longer cable stretches clock low time and requires less than 400 kHz speed operation

# I²C - Bus expander, hub or repeater

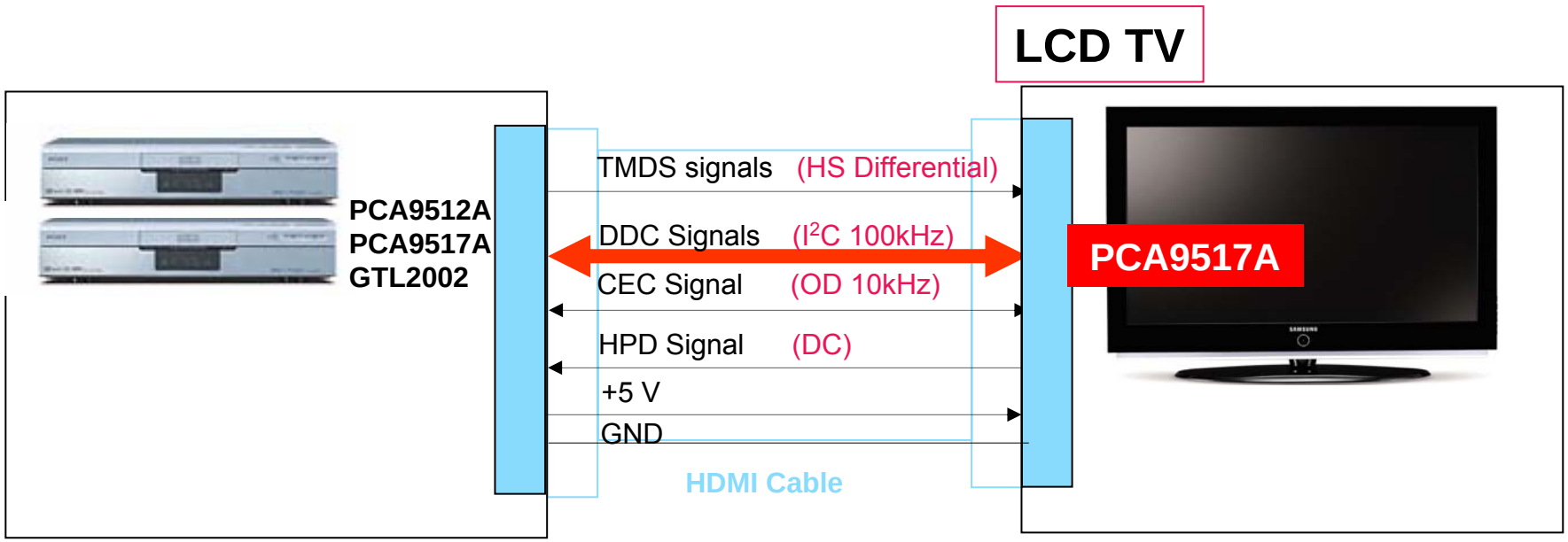
## PCA9517A - HDMI DDC Buffer for LCDTV

Features

- 2-wire bidirectional DDC Buffer
- Dual supply rail for DDC level shift
- Normal I/O swing on A-side
- 5kV ESD HBM

Benefits

- Isolates capacitance and noise and level shift
- Incurs no offset voltage



# I²C - Bus expander, hub or repeater

## PCA9507 HDMI DDC Buffer/Extender

| Features | Benefits |
|---|---|
| Bidirectional bus buffer | Isolate capacitance & noise on DDC line |
| Rise time accelerator on A-side | Extend DDC line > 18 m |
| Strong A-side driver | Increase noise margin on DDC line |
| 5 kV ESD HBM | Require no external discrete components |
| Dual Supplies | Permit  5V DDC level shift |

**DVDR/STB**

**Digital TV**

# I²C - Bus expander, hub or repeater

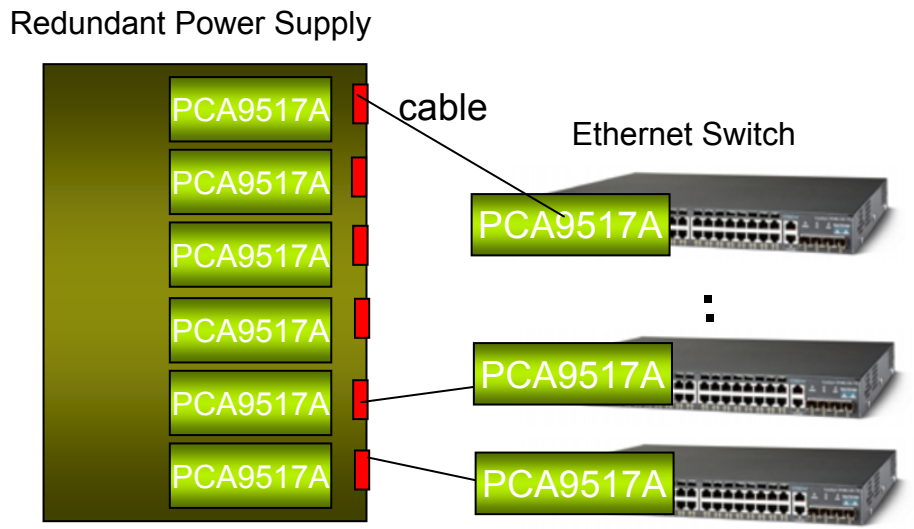
## PCA9517A – High ESD I²C Bus Buffer

Features:

- I²C bidirectional buffer
- Dual supply rails
  - Voltage translation
- Normal I/O voltage swing with high drive on A-side
- 5 kV ESD HBM

Benefits:

- Isolates capacitance and noise for cable application
- Incurs no offset voltage

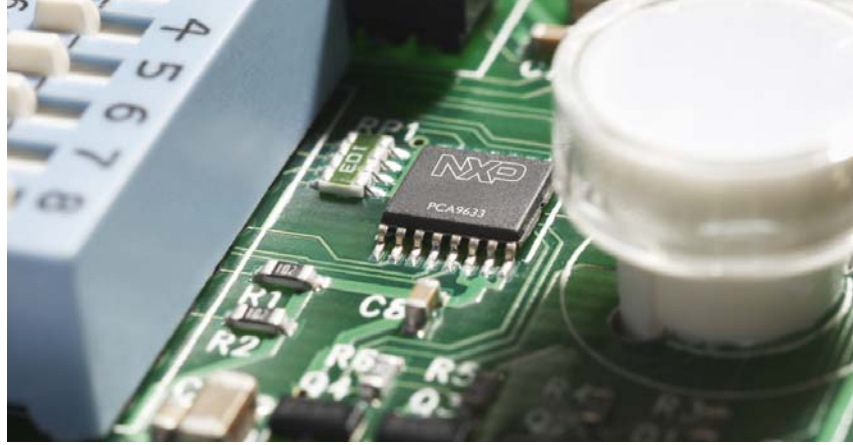



Redundant Power Supply Ethernet Switch
PCA9517A buffers each end of the cable

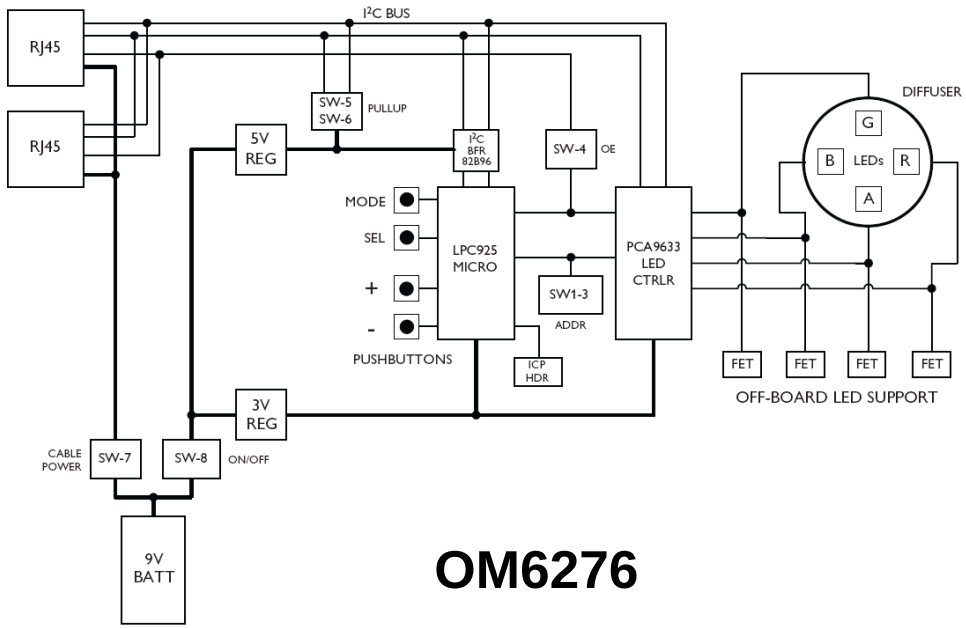
# I²C Demo Boards

# I²C - Demo Boards

## PCA9633 LED dimmer / blinker




- Complete, ready-to-run test environment for dimming and blinking LEDs
- Preprogrammed with six operating modes, including color wash and random color, and can be reprogrammed to create custom color patterns
- The board is configured with four LEDs, one each for red, green, blue, and amber.



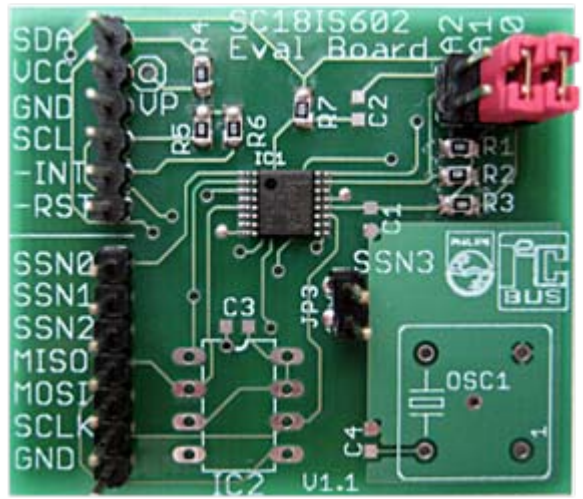

**OM6276**

http://www.standardics.nxp.com/support/boards/pca9633

# I²C - Demo Boards

## SC18IS602 I²C slave to SPI master

- The demo board interfaces to the host processor with an I²C-bus interface
- Any combination of a total of four SPI devices or GPIOs can be controlled.
- The demo board supports up to 8 selectable I²C device addresses via jumpers.
- Can be used for demonstrating:
  - SC18IS602: SPI master with speeds up to 1.8 Mbps using an internal oscillatorI²C-bus slave interfaces with speeds up to 400KHz.
  - SC18IS603: SPI master with speeds up to 4Mbps using an external oscillator. I²C-bus slave interfaces with speeds up to 400KHz.



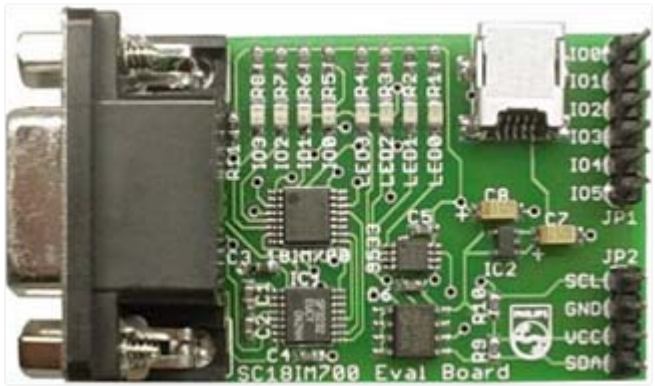

**OM6274**

http://www.standardics.nxp.com/support/boards/sc18is602

# I²C - Demo Boards

## SC18IM700 UART to I²C master / GPIO bridge

- The demo board contains:
  - SC18IM700 UART-to-I2C master/GPIO bridge: controls the operation of the on-board I²C components
  - PCA9533 I²C 4-bit LED dimmer: LEDs connected to it
  - PCF8570 I²C RAM: storing data on demo board
  - Additional LEDs are connected to the GPIO pins of the SC18IM700.



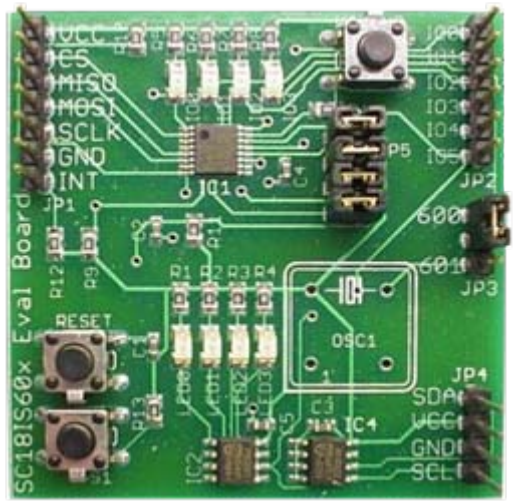

**OM6272**

http://www.standardics.nxp.com/support/boards/sc18im700

# I²C - Demo Boards

## SC18IS600 SPI slave to I²C master / GPIO bridge

- Connects to a host processor via an SPI-bus interface and connects to an I²C slave device via header pins
- The demoboard contains on-board LEDs, which can be controlled by the bridge's programmable GPIO. In addition, the demoboard allows the SPI host to operate an on-board I²C LED dimmer and an on-board I²C EEPROM via the SPI bus interface.
- Board contains:
  - SC18IS600: features SPI speeds up to 1.2 Mbps using an internal oscillator.
- The SC18IS600 on the board can be replaced by:
  - SC18IS601: features SPI speeds up to 3 Mbps using an external oscillator using an external oscillator.



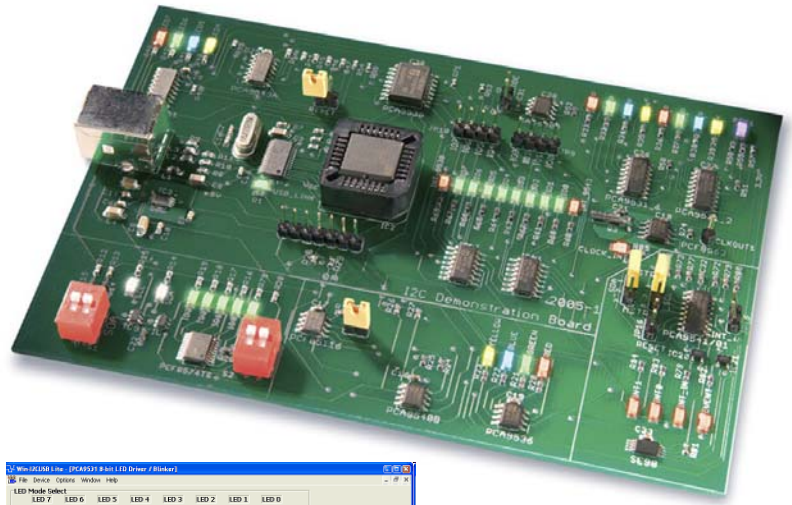

**OM6271**

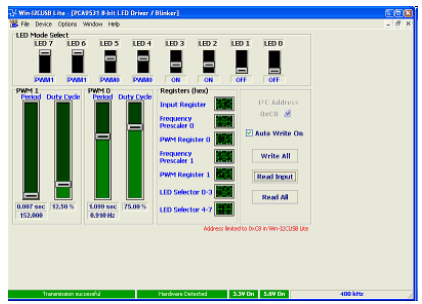
http://www.standardics.nxp.com/support/boards/sc18is600

# I²C - Demo Boards

## I2C 2005-1 kit: easy programming and evaluation of I²C devices

- Board 2005-1 - I²C functions:
  - PCF8574 - GPIO
  - PCA9536 - GPIO
  - PCA9540B - I²C multiplexer
  - PCF85116-3 - EEPROM
  - PCF8563 - real time clock/calendar
  - PCA9538 - GPIO
  - PCA9551 - LED blinker
  - SA56004E - temperature sensor
  - PCA9543A - I²C switch
  - Two PCA9531 - LED dimmer
  - PCA9541/01 - 2-to-1 I²C master selector
  - SE98 - temperature sensor
- Allows easy training / experimenting / testing / demonstration:
  - Field application engineers
  - Designers
  - Educators
- Software control via PC (USB)



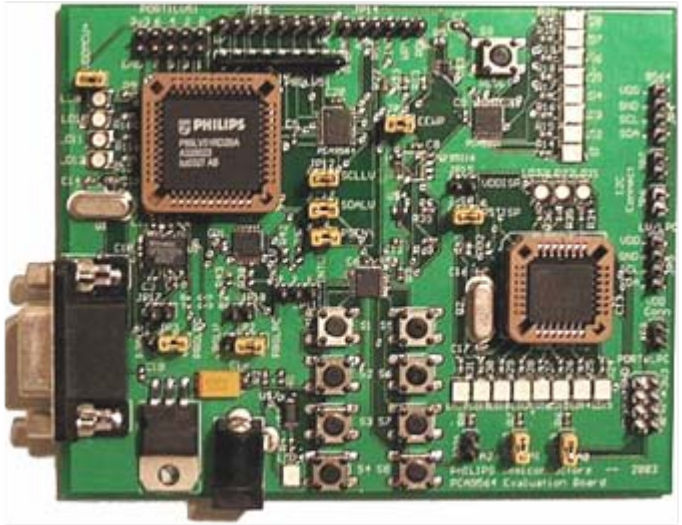
**OM6275**

http://www.standardics.nxp.com/support/boards/i2c20051

# I²C - Demo Boards

## PCA9564: I²C bus controller

- Board demonstrates the I²C bus controller's ability to interface between a master and any master and slave devices connected to its I²C bus.
- Board contains:
  - P89LV51RD2 - microcontroller (connected to PCA9564)
  - PCA9564 - I²C bus controller
  - PCA9531 - 8-bit LED dimmer used as an I²C target slave device
  - P89LPC932 - microcontroller (connected to the I²C bus)
  - PCF85116 - EEPROM
  - PCA9554A - 8-bit GPIO (keyboard readout)
  - Sipex SP3223 - RS-232 transceiver: allows uP devices to be in-system programmed



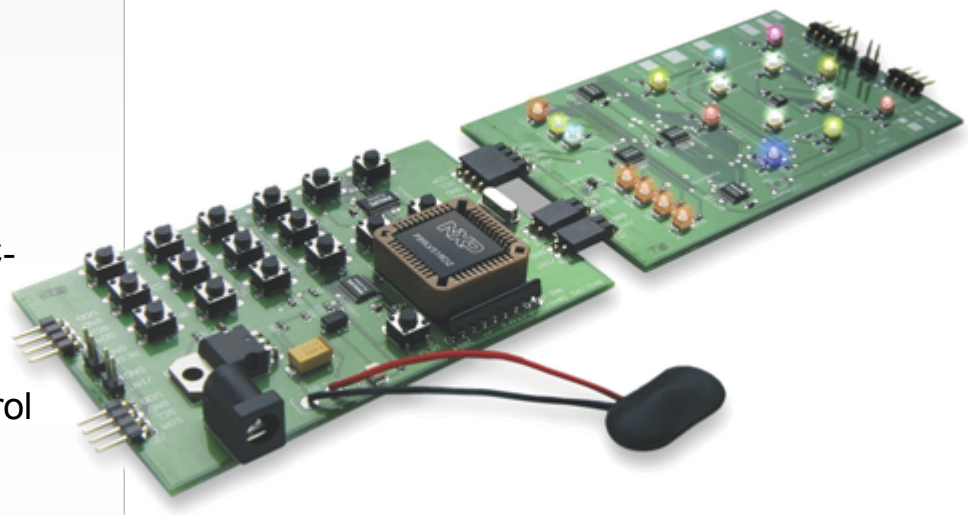
**OM6277**

http://www.standardics.nxp.com/support/boards/pca9564

# I²C - Demo Boards

## LED dimmer board: demonstrates LED lighting and mixing operations

- Two directly connected control cards:
- The LED Control Card contains:
  - LEDs (red, green, blue, white, RGB)
  - Devices used to control LEDs
- The keypad control card contains:
  - NXP P89LV51RD2 microcontroller: interfacing with a NXP PCA9564 I²C-bus controller to generate the I²C commands
  - 16-key keypad: inputs to the microcontroller which allows to control the RGB color mixing, generate fun light patterns, and emulate battery status monitoring
  - Power is supplied by an external 9 V power supply or by a 9 V battery



**OM6279**

http://www.standardics.nxp.com/support/boards/leddemo

# I²C Support Information

# I²C - Support Information

- Internet support (i.e. data sheets, application notes etc.)
  - General: http://www.nxp.com
  - Standard ICs: http://www.standardics.nxp.com
  - I²C logic devices ICs: http://www.standardics.nxp.com/i2c
  - Demo boards: http://www.standardics.nxp.com/support/boards
- Technical Support: i2c.support@nxp.com