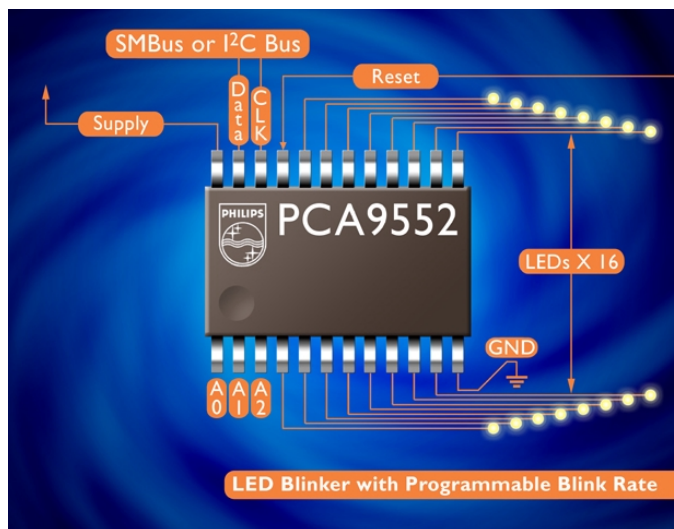


APPLICATION NOTE



AN264 I²C DEVICES for LED DISPLAY CONTROL

PCA9530, PCA9531, PCA9532, PCA9533, PCA9550, PCA9551, PCA9552, PCA9553, SAA1064

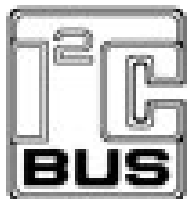
Abstract - Philips Semiconductors family of LED Control Devices: PCA953x LED Dimmers, PCA955x LED Blinkers and SAA1064 LED Digit Display, are detailed in this application note that discusses device operation, software programming, pull up resistor sizing/bus termination and typical applications.

Jean-Marc Irazabal - PCA Technical Marketing Manager

Steve Blozis - PCA International Product Manager

Paul Boogaards - Senior Field Application Engineer

Specialty Logic Product Line
Logic Product Group



Philips Semiconductors

22 July, 2002

TABLE OF CONTENTS

OVERVIEW	3
DESCRIPTION	3
APPLICATIONS	3
PCA95XX LED DIMMERS AND BLINKERS FEATURES	4
<i>Operating Characteristics</i>	4
<i>Device Pinout</i>	5
<i>Ordering Information</i>	5
SAA1064 FEATURES	5
<i>Operating Characteristics</i>	6
<i>Device Pinout</i>	6
<i>Ordering Information</i>	6
DATA SHEETS AND IBIS MODELS	6
TECHNICAL INFORMATION	7
BLOCK DIAGRAM	7
I ² C COMMUNICATIONS	8
HOW TO PROGRAM THE LED DIMMERS AND BLINKERS	9
HOW TO USE UNUSED BITS AS GENERAL PURPOSE I/O.....	13
INTERFACING THE PCA95XX DEVICES WITH LED	13
APPLICATIONS.....	14
BLINKING LEDs	14
DIMMING LEDs.....	19
DIMMING LEDs - ALTERNATIVE SOLUTION USING EXTERNAL OSCILLATOR.....	20
GENERAL PURPOSE I/O.....	21
PCA955X LED BLINKER TYPICAL APPLICATION.....	22
PCA95XX LED DIMMER OR BLINKER USING A CHARGE PUMP AND EXTERNAL DRIVER	22
SAA1064 TYPICAL APPLICATION	23
LED BLINKER EVALUATION BOARD	23
PURPOSE.....	23
EVALUATION BOARD PURCHASE	24
FREQUENTLY ASKED QUESTIONS	24
ADDITIONAL INFORMATION	26
APPENDIX 1 - LED OVERVIEW	26
APPENDIX 2 - LED RESISTOR CALCULATION	26
APPENDIX 3 – RGB COLOR MIXING.....	27

OVERVIEW

Description

The PCA9530, PCA9531, PCA9532 and PCA9533 LED Dimmers and PCA9550, PCA9551, PCA9552 and PCA9553 LED Blinkers control LEDs in I²C and SMBus applications where it is necessary to limit bus traffic or free up the I²C Master's (MCU, MPU, DSP, chipset, etc.) timer. Each LED may be on, off or blinking at one of two programmable rates.

PCA955x (Blinkers): duty cycle is programmable in 256 steps from 0.4% to 100% (LED always on).

PCA953x (Dimmers): duty cycle is programmable in 256 steps from 0% (LED always off) to 99.6%.

The blink rate can vary between a period of 6.25 milliseconds to 1.6 seconds (160 Hz to 0.625 Hz) for the LED Dimmers and 25 milliseconds to 6.4 seconds (40 Hz to 0.156 Hz) for the LED Blinkers. Setting a blink rate higher than 100 Hz and then varying the duty cycle can dim LEDs.

Any bits not used for controlling the LEDs can be used for General Purpose Parallel Input/Output (GPIO) expansion. I/O expansion provides a simple solution when additional I/O is needed for ACPI power switches, sensors, push buttons, alarm monitoring, LEDs, fans, etc. Philips Semiconductors' full line of GPIO devices is detailed in Application Note AN469.

The LED Dimmers and Blinkers are offered in bit widths of 2, 4, 8 and 16-bits. The three hardware pins (A0, A1, A2) on the PCA95x1 and PCA95x2 allow up to eight devices to share the same I²C/SMBus. The single hardware pin (A0) on the PCA95x0 allows up to two devices on the same bus. There are no hardware pins on the PCA95x3 and two versions are offered, PCA95x3-1 and PCA95x3-2 with different fixed addresses allowing the designer to place up to two identical devices on the same bus.

An external active low hardware reset pin (/RESET) is provided on the PCA95x0, PCA95x1 and PCA95x2 to reset the registers to the default state in case the bus locks up, without having to cycle power to the equipment.

The SAA1064 is a bipolar integrated circuit made in an 18 V process. The circuit is especially designed to drive four 7-segment LED displays with decimal point, by means of multiplexing between two pairs of digits. It features an I²C Bus slave transceiver interface with the possibility to program four different slave addresses, a Power Reset flag, 16 current sink outputs controllable by software up to 21 mA, two multiplex drive outputs for common anode segments, an on-chip multiplex oscillator, control bits to select static, dynamic and blank mode, and one bit for segment test. It can also be used to drive up to 32 LEDs at seven different brightness levels.

Applications

These devices can be used for a wide variety of applications:

Blinking LEDs – The PCA95xx LED Dimmers and Blinkers have been designed with an internal oscillator to blink LEDs without continuous commands from the master or tying up a master's timer.

Dimming LEDs – The PCA953x LED Dimmers can dim LEDs using the highest available frequency (160 Hz) and by changing the duty cycle. The SAA1064 allows up to seven levels of brightness control.

LED Segment Control – The SAA1064 can change the indication of two 7-segment displays with period statically and can change four 7-segment displays with period dynamically. The PCA95xx devices can individually change up to sixteen segments.

General Purpose I/O – Bits not used to drive LEDs can be used as normal General Purpose I/O. These are true open drain output devices and a pull-up resistor is needed to set the output in a high state.

PCA95xx LED Dimmers and Blinkers Features

- Designed for operation on the I²C bus, also compatible with SMBus
- Two User Programmable Blink Rates and Duty Cycles
- Blink rate between 6.25 ms and 1.6 s (160 Hz to 0.625 Hz) in 256 adjustable steps for the PCA953x LED Dimmers. Default power-up and reset value set at 160 Hz
- Blink rate between 25 ms and 6.4 s (40 Hz to 0.156 Hz) in 256 adjustable steps for the PCA955x LED Blinkers. Default power-up and reset value set at 0.156 Hz
- Duty cycle between 0.4% and 100% for the PCA955x LED Blinkers. Default power-up and reset value is 50 %
- Duty cycle between 0% and 99.6% for the PCA953x LED Dimmers. Default power-up and reset value is 50 %
- Internal oscillator is accurate to +/- 10% and requires no external components
- Open drain outputs can directly drive LEDs
- Maximum of 25 mA sink per bit at 5.0 V operation
- Maximum device current limits:
 - 2-bit PCA9530/50 = 50 mA
 - 8-bit PCA9531/51 = 100 mA
 - 16-bit PCA9532/52 = 200 mA (100 mA per 8-bit group)
 - 4-bit PCA9533/53 = 100 mA
- Output driver states readable via I²C/SMBus
- Any bit not used to drive an LED can be used as a normal General Purpose I/O expander.
- Active low hardware reset pin (/RESET) (PCA95x0, PCA95x1 and PCA95x2 only) and a built-in Power On Reset (POR) initialize the registers to their default state, causing all the outputs to be Off
- Up to 3 hardware pins (A0, A1, A2) available to change the I²C address and allow up to 8 identical PCA953x or PCA955x devices (2 of the PCA9530/50 and PCA9533/53 or 8 of the PCA9531/32/51/52) to share the same I²C/SMBus.
- Low standby current (I_{stb}) of 1.5 µA max

Important note: The PCA953x Dimmers register default values for frequency/duty cycle and LED ON/OFF control signals are not identical to the PCA955x Blinkers and minor software changes are required when changing from one type to the other.

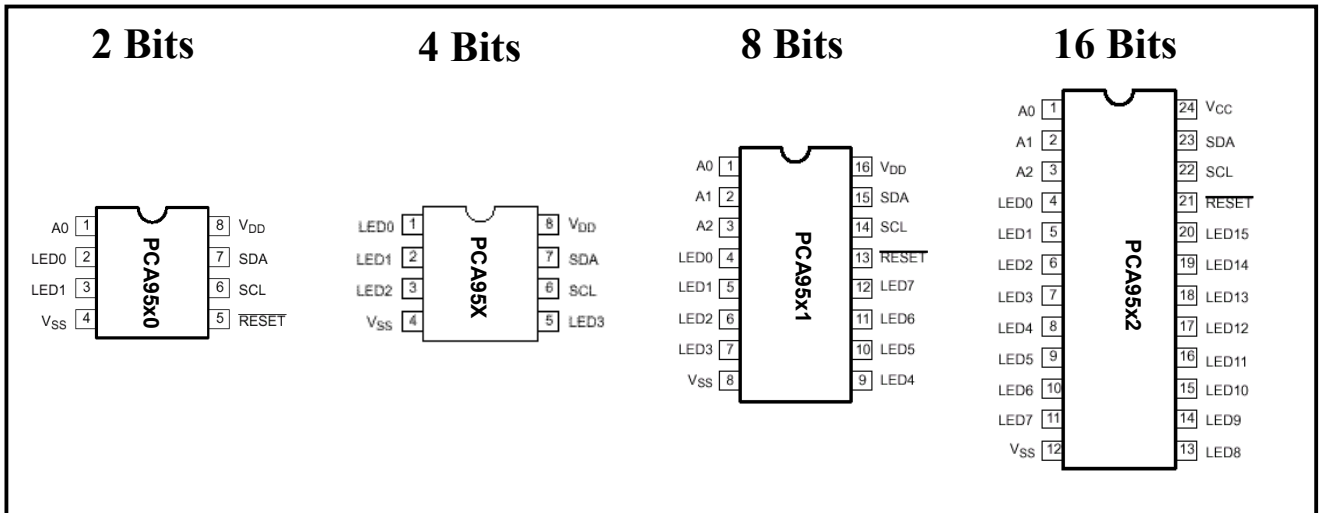
DEVICE NAME	LED BLINKER (Bits)	FEATURES						
		# of I2C ADDRESSES	HARDWARE RESET	PACKAGES				
				PIN COUNT	SO (narrow)	SO (wide)	TSSOP	HVQFN
PCA9530/50	2	2	●	8	D		DP	
PCA9533/53	4	2		8	D		DP	
PCA9531/51	8	8	●	16	D		PW	BS
PCA9532/52	16	8	●	24		D	PW	BS

Operating Characteristics

- I²C and SMBus compatible

- ESD protection exceeds 2000 V HBM per JESD22-A114, 150 V MM per JESD22-A115 and 1000 V CDM per JESD22-C101
- JESDEC Standard JESD78 Latch-up testing exceeds 100 mA
- Manufactured in high-volume CMOS process
- 2.3 V to 5.5 V operating voltage
- 6.0 V tolerant I/Os
- -40 °C to 85 °C operating temperature range
- 0 kHz to 400 kHz clock frequency

Device Pinout



Ordering Information

LED Dimmers

Package	Container	PCA9530	PCA9531	PCA9532	PCA9533-1	PCA9533-2
SO	Tube	PCA9530D	PCA9531D	PCA9532D	PCA9533-1D	PCA9533-2D
	T & R	PCA9530D-T	PCA9531D-T	PCA9532D-T	PCA9533-1D-T	PCA9533-2D-T
TSSOP	Tube	-	PCA9531PW	PCA9532PW	-	-
	T & R	PCA9530DP-T	PCA9531PW-T	PCA9532PW-T	PCA9533-1DP-T	PCA9533-2DP-T
HVQFN	T & R	Not Available	PCA9531BS-T	PCA9532BS-T	-	-

LED Blinkers

Package	Container	PCA9550	PCA9551	PCA9552	PCA9553-1	PCA9553-2
SO	Tube	PCA9550D	PCA9551D	PCA9552D	PCA9553-1D	PCA9553-2D
	T & R	PCA9550D-T	PCA9551D-T	PCA9552D-T	PCA9553-1D-T	PCA9553-2D-T
TSSOP	Tube	-	PCA9551PW	PCA9552PW	-	-
	T & R	PCA9550DP-T	PCA9551PW-T	PCA9552PW-T	PCA9553-1DP-T	PCA9553-2DP-T
HVQFN	T & R	-	PCA9551BS-T	PCA9552BS-T	-	-

SAA1064 Features

- Designed for operation on the I²C bus but compatible with SMBus
- 16-bit output drives four 7-segments displays with decimal or up to 32 LEDs
- Internal oscillator is accurate to +/- 10% and requires no external components
- Open drain outputs can directly drive LEDs
- Maximum of 21 mA sink per segment

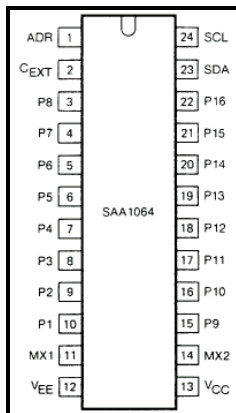
- Output driver states readable via I²C/SMBus
- Standby current (I_{stb}) of 9.5 mA (typical)
- Hardware pin to change the I²C address and allow up to four SAA1064 devices to share the same I²C/SMBus by holding the pin at GND, 1/3 V_{CC}, 2/3 V_{CC} or V_{CC}

DEVICE NAME	LED BLINKER (Bits)	FEATURES				
		# of I2C ADDRESSES	HARDWARE RESET	PACKAGES		
				PIN COUNT	DIP	SO (wide)
SAA1064	16	4		24	P	T

Operating Characteristics

- I²C and SMBus compatible
- 4.5 V to 15.0 V operating voltage
- 5.5 V tolerant I/Os
- -40 °C to 85 °C operating temperature range
- 0 kHz to 100 kHz clock frequency

Device Pinout



Ordering Information

Package	Container	SAA1064
DIP	Tube	SAA1064N
SO	Tube	SAA1064D
	T & R	SAA1064D-T

Data Sheets and IBIS Models

Data sheets and IBIS models (except for SAA1064 IBIS) can be downloaded from www.philipslogic.com/i2c

TECHNICAL INFORMATION

Block Diagram

The PCA9530/31/32/33 LED Dimmers and PCA9550/51/52/53 LED Blinkers are open drain General Purpose I/O, controlled via the I²C bus, with an internal oscillator that allows the devices to independently control and maintain the blink rate of an LED. The block diagram is shown in Figure 1.

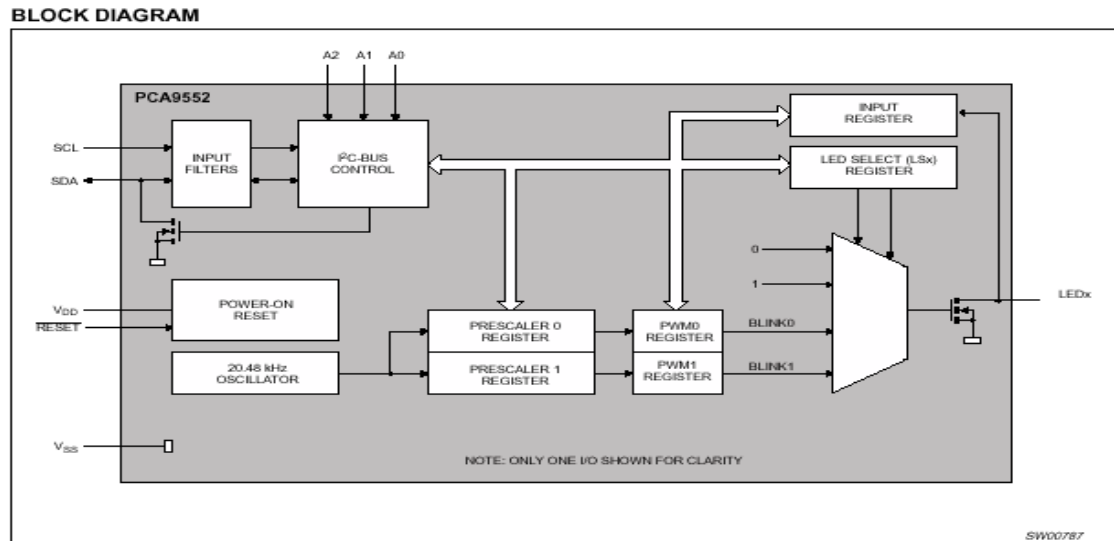


Figure 1. PCA9532/52 Block Diagram

All PCA95xx LED Dimmers and Blinkers offer a built-in Power On Reset (P.O.R.) circuit block, which ensures all LED outputs and I²C I/O's are deactivated while the device is being powered up. The outputs are held in a high impedance state that supports hot insertion.

All LED Dimmers and Blinkers, except the PCA95x3, offer an external hardware pin reset capability. An active low asynchronous reset (/RESET pin) will return all the internal registers to the default state with LED outputs Hi-Z, as well as reset the I²C state machine. This is useful when a rogue device is holding the bus low. It allows the master to reset all devices and regain control of the bus by simply doing the reset without having to cycle power to the equipment and performing a POR to all I²C devices. An external 100 k Ω pull up resistor is required to hold the reset pin high to allow normal operation of the devices. The PCA9533 and PCA9553 do not have this external reset feature due to pincount limitations.

Up to three external hardware address pins are provided for I²C address modification, allowing up to eight PCA95xx devices to operate on the same I²C bus. The address pins must be held high to V_{DD} or low to Ground. It is recommended they be held high through a 10 k Ω pull-up resistor to limit current flow in case of a short but if desired they can be tied directly to V_{DD}.

The PCA95xx devices operate at a supply voltage from 2.3 V to 5.5 V and the SDx/SCx, address and reset pins are all tolerant to voltages up to 6.0V.

I²C Communications

PCA95xx devices support both standard mode (100 kHz) and fast mode (400 kHz) I²C protocols.

A standard I²C communication between a master controller and a PCA95xx device contains the following sequence:

- A Start condition
- An 8-bit word with the following information:
 - a) PCA95xx device addressing. 7 bits (as shown in Table 1) compose the address.
 - b) The 8th bit (Least Significant Bit) is the Read (Logic Level 1) or Write (Logic Level 0) instruction.
- Acknowledge from the slave (PCA95xx addressed device)
- If a Write instruction is requested, the next 8-bit word is the Control register. It contains the address of the Data register the master controller wants to program. Data registers contain the 2 programmable frequency values, the 2 programmable duty cycles and the programmable driver output states (On, Off, Blinking Rate 1, Blinking Rate 2). Addresses for those registers are explained in the Table 2 below.
- After the control register information, the addressed data register will be programmed with the next 8-bit word sent to the PCA95xx device.
- The Auto Increment feature allows writing several consecutive registers. Every time an 8-bit word is sent to the PCA95xx, the address index will be incremented and the next data register will be programmed. Address index will rollover to 0 when the last register has been accessed and programmed.
- If a Read instruction is requested, the bus master becomes a master receiver and the slave PCA95xx device becomes a slave transmitter. Driver outputs logic levels, programmed frequencies and duty cycles and output drivers states can then be read.

The Auto Increment feature allows reading several consecutive registers. Every time the master controller reads a register, the PCA95xx address index will be incremented and the next data register will be read. Address index will rollover to 0 when the last register has been read. **Note that the Read Sequence MUST start by a register different from 0.**
- When the requested operation is a Write, the slave PCA95xx sends an Acknowledge to the master controller after each data byte has been received.
- When the requested operation is a Read, the master controller sends an Acknowledge to the slave PCA95xx device after each data byte has been received.
- A Stop condition.

Device Type	I ² C Address						
PCA9530/50	1	1	0	0	0	0	A0
PCA9531/51	1	1	0	0	A2	A1	A0
PCA9532/52	1	1	0	0	A2	A1	A0
PCA9533/53-1	1	1	0	0	0	1	0
PCA9533/53-2	1	1	0	0	0	1	1

Table 1. PCA953x LED Dimmer and PCA955x LED Blinker Addresses

Note: A2, A1 and A0 , programmable input pins allow the use of up to two PCA9530/50 and eight PCA9531/32/51/52 devices on the same I²C bus. PCA9533 and PCA9553 only have a fixed assigned address. Two different fixed addresses are available (PCA95x3-1 and PCA95x3-2) and allow the use of 2 same devices on the same bus in addition to 2 of the PCA9530/50 devices.

The Control register determines which data register needs to be addressed (Read or Write operation).

- Bit [0:2] (PCA9530/31/33/50/51/53) and Bit [0:3] (PCA9532/52) is the bit used to address a specific register.
- Bit [4] is the Auto-Increment option and allows optimized I²C programming.
 - When this bit is at Logic Level 0, the Auto-Increment option is disabled
 - When this bit is at Logic Level 1, the Auto-Increment option is enabled

7	6	5	4	3	2	1	0	PCA9530/33/50/53	PCA9531/51	PCA9532/52
0	0	0	x	0	0	0	0	Input Register	Input Register	Input Register 0
0	0	0	x	0	0	0	1	Frequency Prescaler 0	Frequency Prescaler 0	Input Register 1
0	0	0	x	0	0	1	0	PWM Register 0	PWM Register 0	Frequency Prescaler 0
0	0	0	x	0	0	1	1	Frequency Prescaler 1	Frequency Prescaler 1	PWM Register 0
0	0	0	x	0	1	0	0	PWM Register 1	PWM Register 1	Frequency Prescaler 1
0	0	0	x	0	1	0	1	LED Selector (0 and 1)	LED Selector 0 (0 to 3)	PWM Register 1
0	0	0	x	0	1	1	0	Not allowed	LED Selector 1 (4 to 7)	LED Selector 0 (0 to 3)
0	0	0	x	0	1	1	1	Not allowed	Not allowed	LED Selector 1 (4 to 7)
0	0	0	x	1	0	0	0	Not allowed	Not allowed	LED Selector 2 (8 to 11)
0	0	0	x	1	0	0	1	Not allowed	Not allowed	LED Selector 3 (12 to 15)
x	x	1	x	x	x	x	x	Not allowed	Not allowed	Not allowed
x	1	x	x	x	x	x	x	Not allowed ^(*)	Not allowed ^(*)	Not allowed ^(*)
1	x	x	x	x	x	x	x	Not allowed	Not allowed	Not allowed

Table 2. Data Register Mapping

White means that the registers are Read Only Registers.
 Light gray means that the registers are Read/Write Registers.
 Dark gray indicates the not allowed combinations.

(*): Except “01000000” for the Bypass Mode. For additional information, see Paragraph “Dimming LEDs – Alternative solution using external oscillator”, Chapter “APPLICATIONS”.

How to Program the LED Dimmers and Blinkers

- **Blinking Rates.**
 Four parameters (PSC0, PWM0, PSC1 and PWM2) define the 2 programmable blink rates, BLINKING RATE 1 and BLINKING RATE 2 .
 - BLINKING RATE 1 is determined by the set of values PSC0 and PWM0
 - BLINKING RATE 2 is determined by the set of values PSC1 and PWM1
 - PSC0 and PSC1 values determine the blinking frequency and are programmed by writing data in the registers “Frequency Prescaler 0” and “Frequency Prescaler 1”.
 The frequency can be programmed between 160 Hz (period = 6.25 ms) and 0.625 Hz (period = 1.6 s) for the LED Dimmers and 40 Hz (period = 25 ms) and 0.156 Hz (period = 6.4 s) for the LED Blinkers in 256 steps.
 - PWM0 and PWM1 values determine the duty cycle, which is the ratio between the ON time and the period and are programmed by writing data in the registers “PWM Register 0” and “PWM Register 1”.
 The blinking rate (duty cycle) can be programmed between 100% (default) and 0.4 % in 256 steps for the Blinkers (PCA955x) and 0% (default) to 99.6 % for the Dimmers (PCA953x).
- Figure 2 gives more information on those 4 parameters.

Each parameter is an 8-bit word and can vary from 00_h to FF_h (0 to 255 in decimal).
 The register addressing is explained in Table 2.

Example using the PCA955x devices:

		Decimal value	Hexadecimal Value
BLINKING RATE 1 = 1 Hz, 50 % duty cycle	PSC0	37	25
	PWM0	128	80
BLINKING RATE 2 = 4 Hz, 25 % duty cycle	PSC1	9	09
	PWM1	192	E4

Table 3. Data Register Mapping

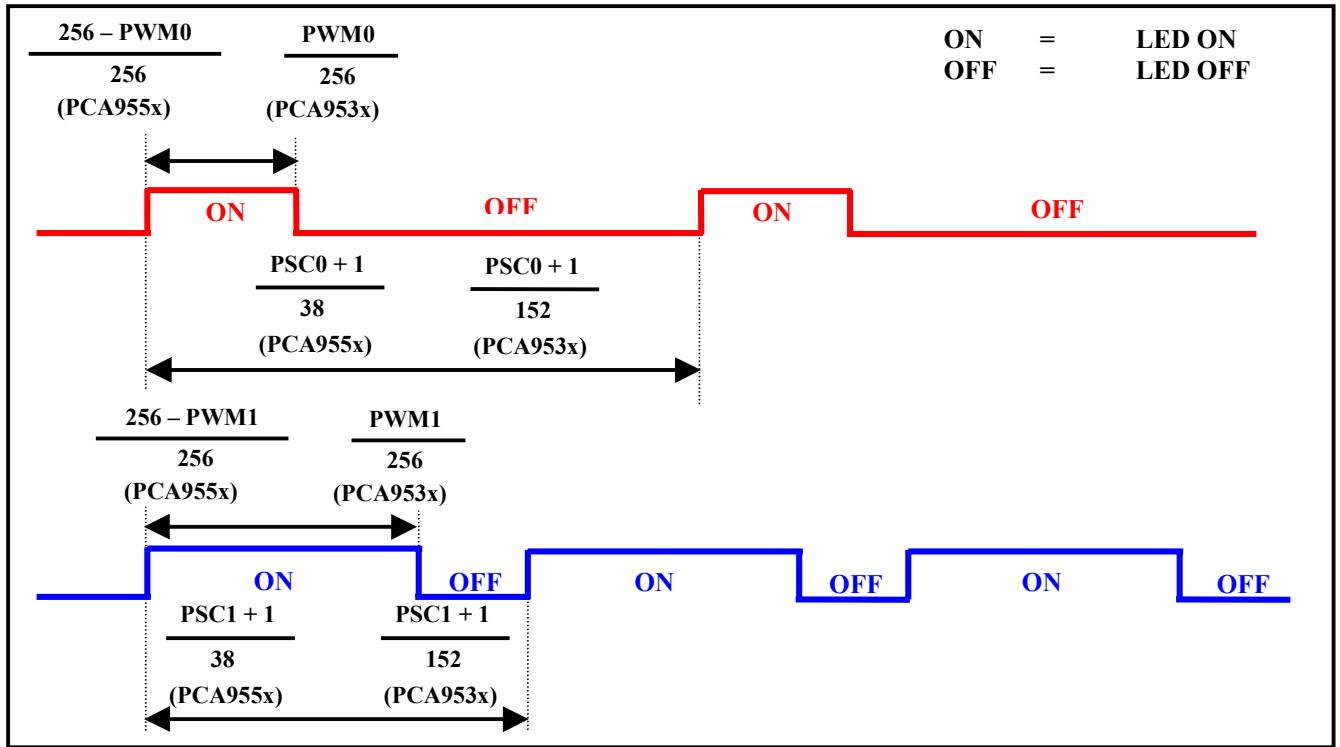


Figure 2. Blinking Waveforms

Note: the waveforms above represent the LED ON and LED OFF states.

A "LED ON" state means that the LEDx pin logic level is "0" (Ground)

A "LED OFF" state means that the LEDx pin logic level is "Hi-Z" (Floating)

- Output Driver States

Once the 4 values have been programmed, the output driver state can be chosen between the 2 predefined blinking rate, a permanent ON state or a permanent OFF state.

This is done by writing the LED Selector register(s). The register addressing is explained in Table 2.

Table 3 gives mapping information for each LED Selector Register.

		7	6	5	4	3	2	1	0
PCA9530/50	LED Selector	Not Used		Not Used		LED1		LED0	
PCA9531/51	LED Selector 0	LED3		LED2		LED1		LED0	
	LED Selector 1	LED7		LED6		LED5		LED4	
PCA9532/52	LED Selector 0	LED3		LED2		LED1		LED0	
	LED Selector 1	LED7		LED6		LED5		LED4	
	LED Selector 2	LED11		LED10		LED9		LED8	
	LED Selector 3	LED15		LED14		LED13		LED12	
PCA9533/53	LED Selector	LED3		LED2		LED1		LED0	

Table 4. LED Selector Registers Mapping

Table 4 and 5 explain the different Output driver states.

LEDx		Output Driver State
0	0	Output logic level = Low – LED permanently ON
0	1	Output logic level = Hi-Z – LED permanently OFF
1	0	Output blinking at BLINKING RATE 1
1	1	Output blinking at BLINKING RATE 2

Table 5. Output Drivers States for PCA955x LED Blinkers

MSB	LSB	Output Driver State
0	0	Output logic level = Hi-Z – LED permanently OFF
0	1	Output logic level = Low – LED permanently ON
1	0	Output blinking at BLINKING RATE 1
1	1	Output blinking at BLINKING RATE 2

Table 6. Output Drivers States for PCA953x LED Dimmers

- Register differences between PCA953x Dimmers and PCA955x Blinkers
Default values are different between the PCA955x Blinkers and the PCA953x Dimmers. Some software changes need to be done in case of type replacement. Differences are listed below in Table 6 and 7.

	PSC Registers Default State		Permanently ON Command		Permanently OFF State Default State	
PCA953x	00 _h	160 Hz	0	1	0	0
PCA955x	FF _h	0.156 Hz	0	0	0	1

Table 7. PCA953x / PCA955x register differences

	Frequency (Hz)			Duty Cycle (%)		
	00 _h	...	FF _h	00 _h	...	FF _h
PCA953x	160	...	0.625	0	...	99.6
PCA955x	40	...	0.156	100	...	0.4

Table 8. PCA953x / PCA955x frequency and duty cycle differences

Note: Duty Cycle is defined as the ratio between the LED ON time and the total period (ON + OFF time).
For the PCA955x, LED can be ON all the time (100%) or almost OFF all the time (0.4%).
For the PCA953x, LED can be OFF all the time (0.0%) or almost ON all the time (99.6%).

- Auto-Increment Option: Write to the registers
 - The AI (Auto Increment) bit in the Control Register (Bit #4) allows the user to program the device by using a minimum number of I²C commands. When set to Logic Level 0, the feature cannot be used. When set to Logic Level 1, the feature is enabled.
 - Blinking Rate 1 & 2 and LED Selector register(s) can be programmed once and will remain valid until a new blinking rate and/or a new output driver state is required.

Without the Auto Increment option, to program all the registers (example given for the PCA9550 below), the I²C command would be the following (dark gray (blue) means that the instruction is generated by the master controller, light gray (yellow) means that the instruction is generated by the slave device):

S	PCA9550 Address + Write	A	Control Register - Reg#1	A	Data Reg#1	A
S	PCA9550 Address + Write	A	Control Register - Reg#2	A	Data Reg#2	A
S	PCA9550 Address + Write	A	Control Register - Reg#3	A	Data Reg#3	A
S	PCA9550 Address + Write	A	Control Register - Reg#4	A	Data Reg#4	A
S	PCA9550 Address + Write	A	Control Register - Reg#5	A	Data Reg#5	A P

Reg#1 is the Frequency Prescaler 0 Register (PSC0).

Reg#2 is the PWM Register 0 (PWM0).

Reg#3 is the Frequency Prescaler 1 Register (PSC1).

Reg#4 is the PWM Register 1 (PWM1).

Reg#5 is the LED Selector Register (LS0).

S is the Start Command or the Re-Start Command.

P is the Stop Command.

A is the Acknowledge sent by the addressed slave PCA9550 device.

With the Auto Increment Option, to program all the registers (example given for the PCA9550 below), the I²C command is the following (dark gray (blue) means that the instruction is generated by the master controller, light gray (yellow) means that the instruction is generated by the slave device):

S	PCA9550 Address + Write	A	Control Register-Reg#1 - AI=1	A	Data Reg#1	A
	Data Reg#2	A	Data Reg#3	A	Data Reg#4	A
	Data Reg#5	A	P			

When the Auto-Increment option is set (AI=1), the internal index counter (counter initialized to 1 in our example above) will be incremented after each byte is sent to the PCA9550.

Note: When the index counter reaches its maximum values (5 for the PCA9530/50 and PCA9533/53, 6 for the PCA9531/51, 9 for the PCA9532/52), it will rollover to 0 and overwrite the previous data if the master controller sends another 8-bit word.

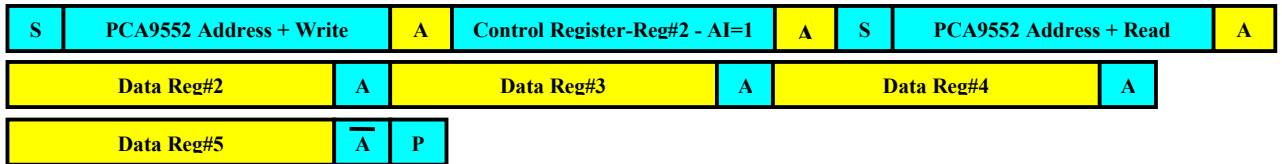
- Auto-Increment Option: Read from the registers

This feature can also be used in a Read mode and may be used to optimize a Reading sequence.

Without the Auto Increment Option, to read the PSC0, PWM0, PSC1, PWM1 programmed values (Example given for the PCA9552 below), the I²C command would be the following (dark gray (blue) means that the instruction is generated by the master controller, light gray (yellow) means that the instruction is generated by the slave device):

S	PCA9552 Address + Write	A	Control Register - Reg#2	A	S	PCA9552 Address + Read	A
	Data Reg#2	\overline{A}	S	PCA9552 Address + Write	A	Control Register - Reg#3	A
S	PCA9552 Address + Read	A	Data Reg#3	\overline{A}	S	PCA9552 Address + Write	A
	Control Register - Reg#4	A	S	PCA9552 Address + Read	A	Data Reg#4	\overline{A}
S	PCA9552 Address + Write	A	Control Register - Reg#5	A	S	PCA9552 Address + Read	A
A	Data Reg#5	\overline{A}	P				

With the Auto Increment Option, to read the same 4 registers (Example given for the PCA9552 below), the I²C command is the following (dark gray (blue) means that the instruction is generated by the master controller, light gray (yellow) means that the instruction is generated by the slave device):



When the Auto-Increment option is set (AI=1), the internal index counter (counter initialized to 1 in our example above) will be incremented after each 8-bit word Data sent by the PCA9552. The next 8-bit word Data sent by the PCA9552 will automatically come from the following Data register.

Note: When the index counter reaches its maximum values (5 for the PCA9530/50 and PCA9533/53, 6 for the PCA9531/51, 9 for the PCA9532/52), it will rollover to 0 if another 8-bit word is sent by the master controller.

Important Note: When a Read sequence is initiated and the AI bit is set to Logic Level 1, the Read Sequence **MUST** start by a register different from 0.

How to use unused bits as General Purpose I/O

Output drivers can also be used as General Purpose Input/Outputs (GPIOs) if they are not used to blink LEDs.

- Use as output:
The PCA95xx outputs have an Open Drain Output structure. A pull-up to V_{DD} is then required if the Output pin has to drive an external device.
Program the 2 bits in the corresponding LED Selector Register as following:

	PCA953x	PCA955x
Logic Level Low	01	00
Logic Level High	00	01

Table 9. Output drivers used as General Purpose Outputs

- Use as input:
Program the 2-bit word in the corresponding LED Selector Register at 00 (PCA953x) or 01 (PCA955x) to disable the Output Open Drain device.
A Logic Level High or Low can then be applied to the corresponding pin.
A master controller can then the read logic Level by requesting a Read on the corresponding Input Register (see Data Register mapping in Table 2)

Interfacing the PCA95xx devices with LED

LED driver output stages have an open drain structure and have the following characteristics:

- 6.0 V tolerant
- 25 mA maximum sinking current capability @ 0.7 V drop and 5.5 V V_{DD}

If the application is within the limits described above, the LED(s) can then be connected directly to the Philips PCA95xx devices. If more than 25 mA are required and/or more than 6.0V needs to be applied to the PCA95xx LED Dimmers or Blinkers I/O, an additional stage needs to be added. A simple PNP transistor withstanding the application requirements can be used to make such an interface, as shown in Figure 3.

R_1 needs to be calculated upon the LED current requirement and V_{DD} .

R_3, R_4, \dots to R_n and R_x are function of the LED characteristics and the brightness requirements.

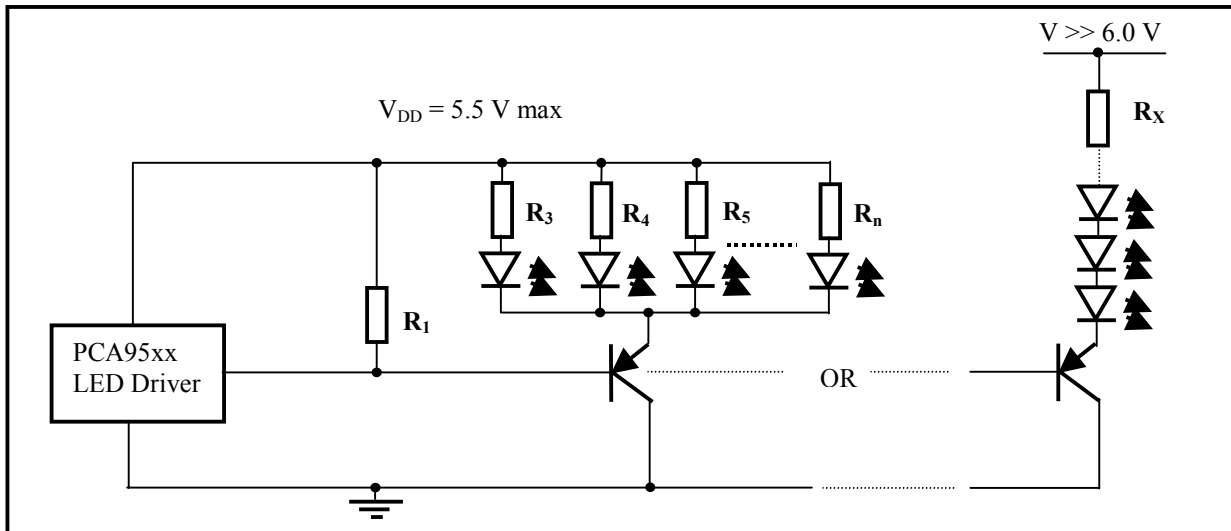


Figure 3. Interfacing the PCA95xx output drivers

APPLICATIONS

Blinking LEDs

The large blinking frequency range makes the PCA955x LED Blinkers suitable to generate different visual messages depending on specific situations. A typical example could be an application where LEDs are ON or blink at a very low speed when no warning or alarm is present. When such a warning or alarm is detected by the system, LEDs can be programmed to blink at a faster speed to catch the attention of an operator or to show a specific problem. Another application can be a cellular phone with LEDs blinking at slow speed when in standby mode and blinking faster when an incoming call is detected.

The PCA953x LED Dimmer devices can also be used to blink LEDs in addition of dim LEDs, but the maximum blink period is limited to 1.6 seconds. If blink rates higher than 1.6 seconds are required (e.g., status indication with frequency of 6 seconds with a short 4% duty cycle), then the master would have to send a command to the device to turn the LED on for 240 ms and off for the 5.76 seconds as it would for a normal GPIO.

- Example 1: 2-alarm system
A very easy way to program a 2-alarm system can be done by using a PCA9550 that blinks 2 LEDs at slow speed when the system is in normal mode. When Alarm 1 and/or Alarm 2 is generated by an external device or condition, the master will make LED₀ blink (if Alarm 1 detected) and/or LED₁ (if Alarm 2 is detected) faster until Alarm 1 and/or Alarm 2 is cleared.

A possible implementation of such system is described in Figure 3.

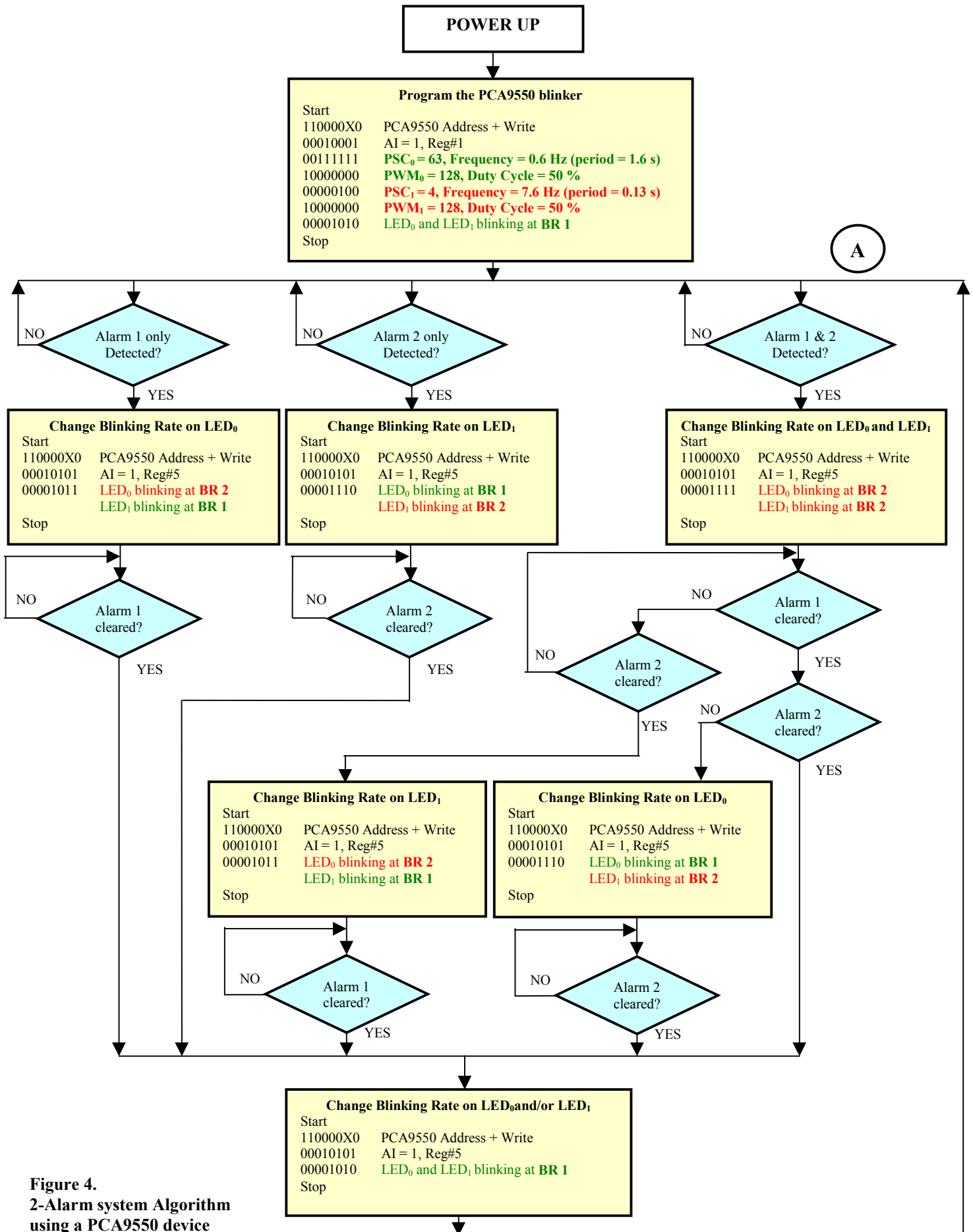


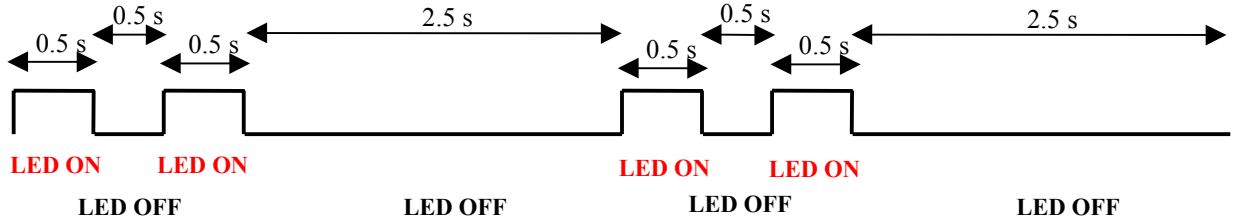
Figure 4.
2-Alarm system Algorithm
using a PCA9550 device

Note:

- BR 1 stands for Blinking Rate 1 (Green in the algorithm) and represents a slow blinking (no Alarm condition)
- BR 2 stands for Blinking Rate 2 (Red in the algorithm) and represents a fast blinking (Alarm condition)
- If a new Alarm condition occurs (in addition of an already existing one), the algorithm will go to the entry point A immediately.

- Example 2: a more complex blinking scheme

In this example, we show how we can program a “non regular” blinking sequence, in others words, a sequence where the LED control waveform is not a simple square voltage but follows the requirements below.



One simple way to do such blinking sequence is to use one of the LED output drivers to generate the blinking signal and another one to generate a masking signal disabling the blinking. The 2 output driver signals need then to be mixed to create the required LED control signal.

Blinking signal will be as following (I/O1):

- Frequency = 1 Hz (Period = 1 Hz)
- Duty Cycle = 50 %



Masking signal will be as following (I/O0):

- Frequency = 0.25 Hz (Period = 4 s)
- Duty Cycle = 50 %



Schematic using a PCA9550 and the code for this application are shown in Figure 5.

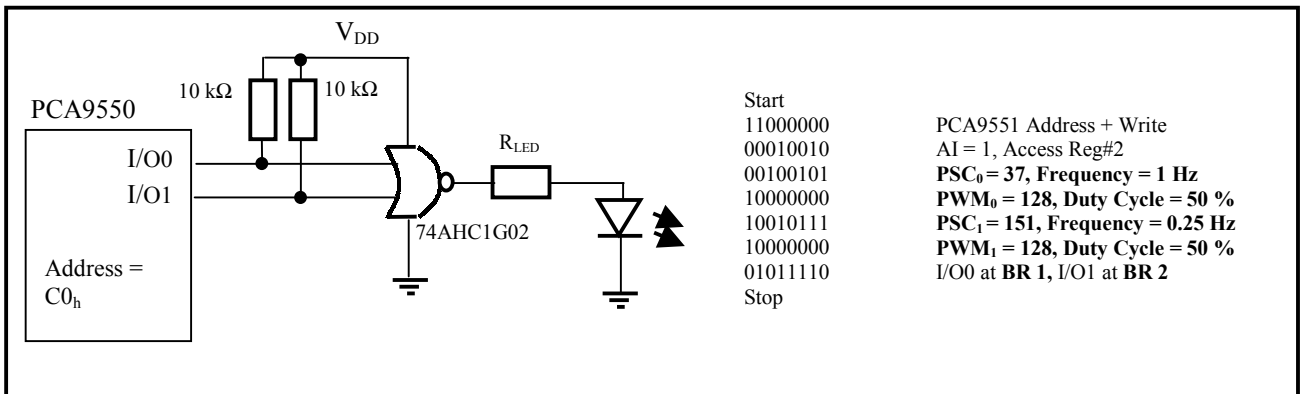


Figure 5. Blinking implementation and I²C code

A NOR gate (e.g., Philips 74AHC1G02 PicoGate) is used to create the required LED control signal. The output of the gate controls directly the LED. Pull-up resistors to V_{DD} are required for I/O0 and I/O1 because of the open-drain structure used for the PCA955x LED blinkers.

R_{LED} is calculated upon the LED characteristics and current requirements.

Important note: There is no synchronization of the I/Os when a change in the LED Selector(s) is done (falling and rising edges may occur at different moments) and some transient state (visible or not to a human eye) may occur at the output of the NOR gate.

- Example 3: How to drive Dual Color LEDs
2 different types of Dual Color LEDs are available in the market. They are described in the Figure 6.

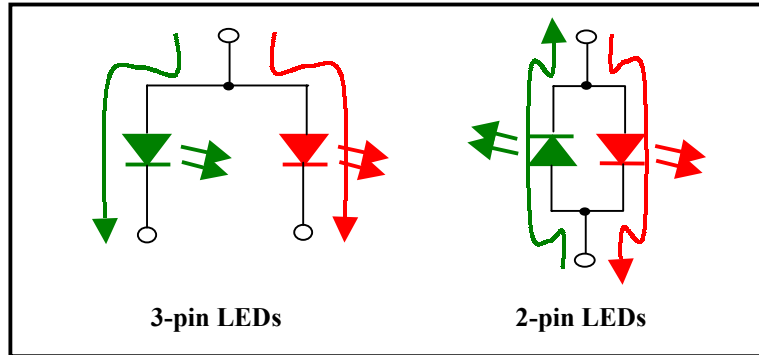


Figure 6. Dual Color LEDs

The 3-pin type devices have a common Anode and separate Cathode. Current flows in the same direction for both devices and can be both ON at the same time.

The 2-pin type devices are in a back to back configuration. Current flows in 2 different directions and cannot be both ON at the same time.

a) 2-pin device control

Since current control needs to be done in 2 different directions and since the Philips PCA955x/PCA953x devices have only a sinking capability (open drain structure), 4 control signals (4 output drivers) are required to command this type of LED.

For both directions, one output will control the blinking (ON/OFF states) and 3 outputs are used to control the current direction.

The solution also requires 2 external switches (e.g., Philips 74HC1G66 picogates) as shown in Figure 7.

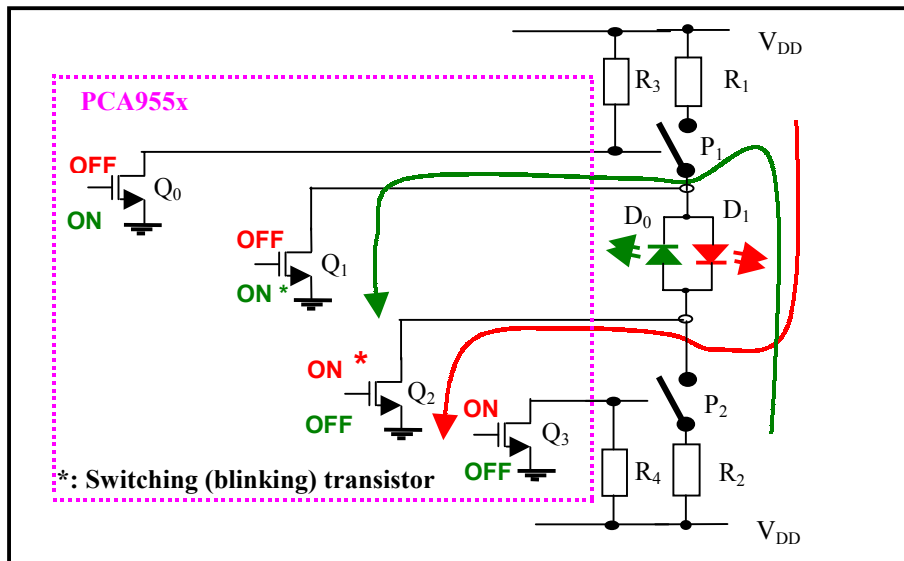


Figure 7. Proposed solution using a PCA955x device - 2-pin Dual Color LED driving

R_1 and R_2 control the current flowing through the LEDs and are calculated upon the LED characteristics and the current requirements.

R_3 and R_4 are pull-up resistors required (due to the open drain structure) and are typically equal to 10 k Ω .

P_1 and P_2 are Philips 74LVC1G66 or 74HC1G66 Bilateral Switches.

D_0 and D_1 are the 2 LEDs in the 2-pin Dual Color LED.

Q_0 to Q_3 are the internal NMOS transistors in the PCA955x LED blinker.

To blink the Green LED (D_0):

- Q_0 needs to be permanently ON (P1 open – command signal = 0)
- Q_1 needs to be turned ON/OFF at the desired frequency/duty cycle
- Q_2 needs to be permanently OFF
- Q_3 needs to be permanently OFF (P2 closed – command signal = 1)

If we use LED[0:3], the corresponding LED selector register needs to be programmed as following:

- 01011000 if the LED blinks at Blinking Rate 1
- 01011100 if the LED blinks at Blinking Rate 2

To blink the Red LED (D_1):

- Q_0 needs to be permanently OFF (P1 closed – command signal = 1)
- Q_1 needs to be permanently OFF
- Q_2 needs to be turned ON/OFF at the desired frequency/duty cycle
- Q_3 needs to be permanently ON (P2 open – command signal = 0)

If we use LED[0:3], the corresponding LED selector register needs to be programmed as following:

- 00100101 if the LED blinks at Blinking Rate 1
- 00110101 if the LED blinks at Blinking Rate 2

b) 3-pin device control

Since current control is in the same direction, the implementation for a 3-pin type Dual Color LED is much simpler than the 2-pin type device and is the same as a discrete 2-LED solution as shown in Figure 8.

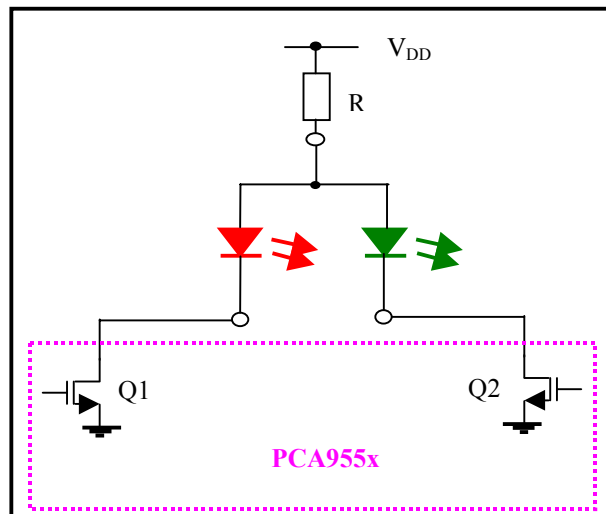


Figure 8. Proposed solution using a PCA955x device - 3-pin Dual Color LED

R controls the current flowing through the LEDs and must be calculated upon the LED characteristics and the current requirements.

Dimming LEDs

The PCA95xx devices can be used to dim LEDs by changing the duty cycle (i.e., ON over ON+OFF ratio) that will change the average current flowing through the LEDs. The bigger the ON time, the brighter the LED will be. To dim the LEDs, set the frequency to the maximum and then use the duty cycle register to control the brightness of the LED. It has been determined that a frequency above 100 Hz is most effective so that the normal eye cannot see the blinking which limits the use of the PCA955x LED Blinkers in this application (maximum programmable blinking frequency is 40 Hz).

Change the duty cycle over time allows applications like:

- Fade LED lighting (soft power on phase, min to max brightness in few seconds)
- Auto regulated lighting depending on the ambient luminosity (LED is dimmed if in a dark environment)
- Red, blue and green LED color mixing applications (see appendix 3)

The example in Figure 9 shows an application where a progressive power on cycle is required. The algorithm explains how to control a PCA9532 driving a 4x4 LED matrix from the OFF state to a max 99.6% brightness in 5s with 5 different step (20%, 40%, 60%, 80% and 99.6%).

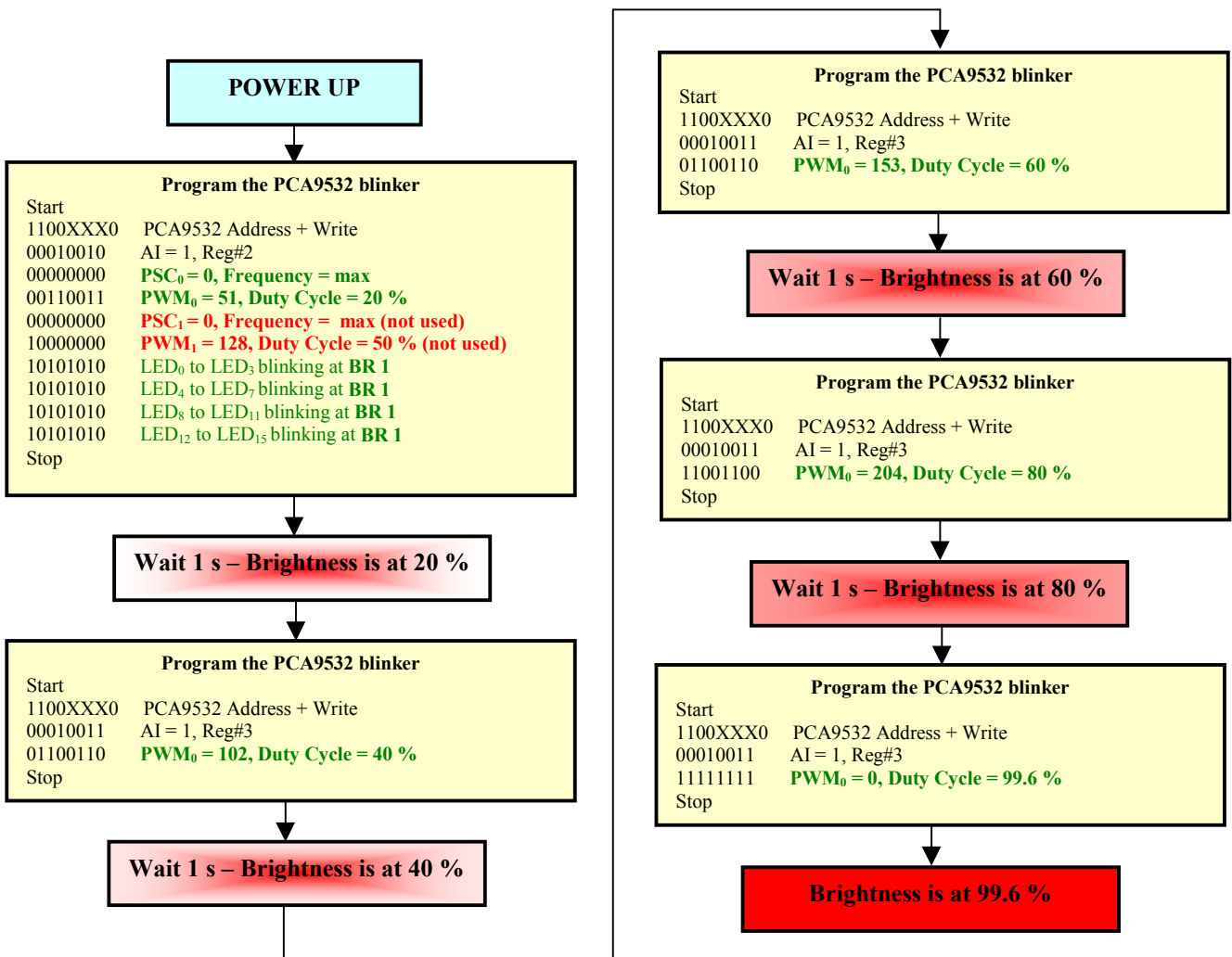


Figure 9. Progressive Brightness control using the PCA9532 device

Dimming LEDs - Alternative solution using external oscillator

The PCA955x maximum blinking frequency is about 40 Hz, which may not be fast enough for some specific applications where a small flicker is not acceptable. In order to be able to use these devices and the PCA953x with a blinking frequency higher than the maximum internal frequency, it's possible to bypass the internal oscillator and apply an external clock instead. The higher frequency external clock can also be used in applications like PWM motor control or tone generation (PWM output tied to piezo-electrical crystal or speaker) where higher frequency outputs are required.

Note, there are some limitations when this mode is used:

- Some glue logic is required (1 AND gate or 1 NAND gate depending on the required programmable hardware pin value).
- An additional control pin is required (from the master controller).
- PCA9530/50 cannot support this bypass mode.
- Output drivers LED0 and LED1 cannot be used with this bypass mode.
- PCA95xx cannot be addressed when the Bypass mode is active. The master must go back to the normal mode to address the slave when this is required.
- Maximum frequency input is dependent on the maximum required frequency at the output driver. Tone generator or fan controller applications may require maximum output frequencies in the 10 to 25 kHz range, thus requiring a maximum input frequency in the 5.1 to 12.8 MHz range.

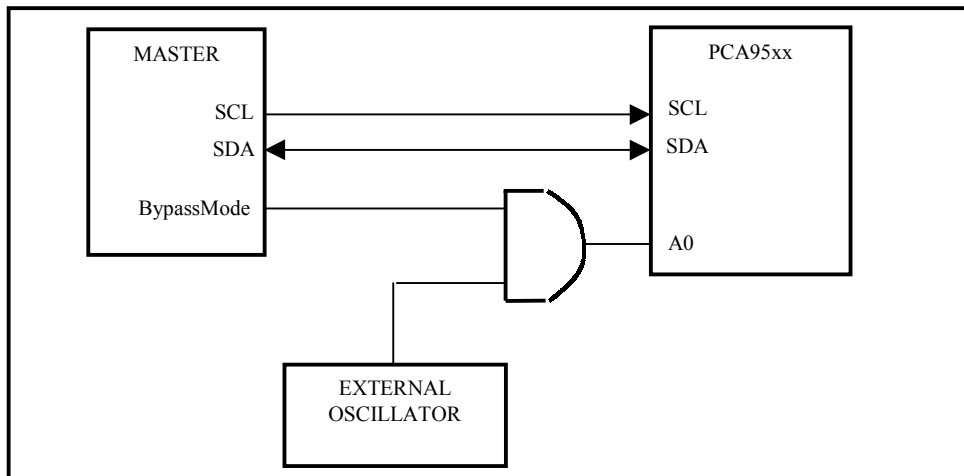


Figure 10. Solution a) – A0 needs to be programmed at Logic Level Low

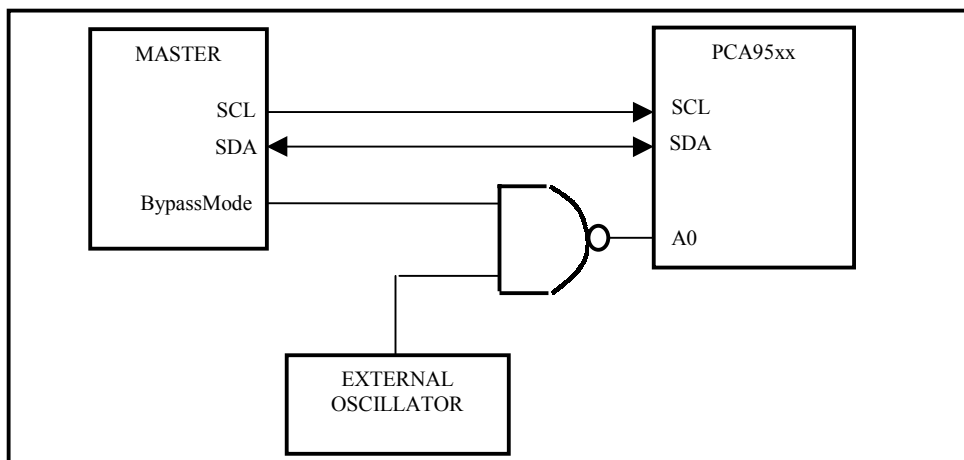


Figure 11. Solution b) – A0 needs to be programmed at Logic Level High

Bypass mode is generated in 2 sequences:

- The master controller makes a Bypass mode request by addressing the targeted PCA95xx device, writing 01000000 (40_h) in the Control register and generating a STOP command.
- The master controller switches to the bypass mode through the Bypass Mode control bit.

Bypass Mode control bit is controlled by the master control and is described as following:

- BypassMode = 0: Normal mode, internal oscillator clock used (A0 = 0 in solution a), A0=1 in solution b).
- BypassMode = 1: Bypass mode, external clock applied to the PCA95xx device

When the Bypass mode is active, the output driver frequency follows the equation:

$$\text{Output Driver Frequency} = \frac{\text{External Clock Frequency}}{512 (\text{PSCx} + 1)}$$

Where PSCx is the decimal value programmed in the Frequency Prescaler x (x = 0 or 1).

Mode of operation

1. Power-up the PCA95xx device with Normal mode active (BypassMode = 0)
2. In normal mode (BypassMode = 0), the PCA95xx device can be addressed and can be programmed (PSC0, PSC1, PWM0, PWM1, output driver states).
3. Switch to Bypass mode (I²C Bypass mode request and then BypassMode = 1). The external clock will be applied to pin A0 and the output driver frequencies will follow the equation above.
4. If an access to the PCA95xx is required (read registers, reprogram the device), the master controller must go to Normal mode first (BypassMode = 0), generate the required I²C commands and go back to Bypass mode (I²C Bypass mode request and then BypassMode = 1). During this time, the external clock will not be applied to the PCA95xx anymore but the access can be short enough not to be seen by a human eye.
5. Go through step 4 every time an access to the PCA95xx is required.

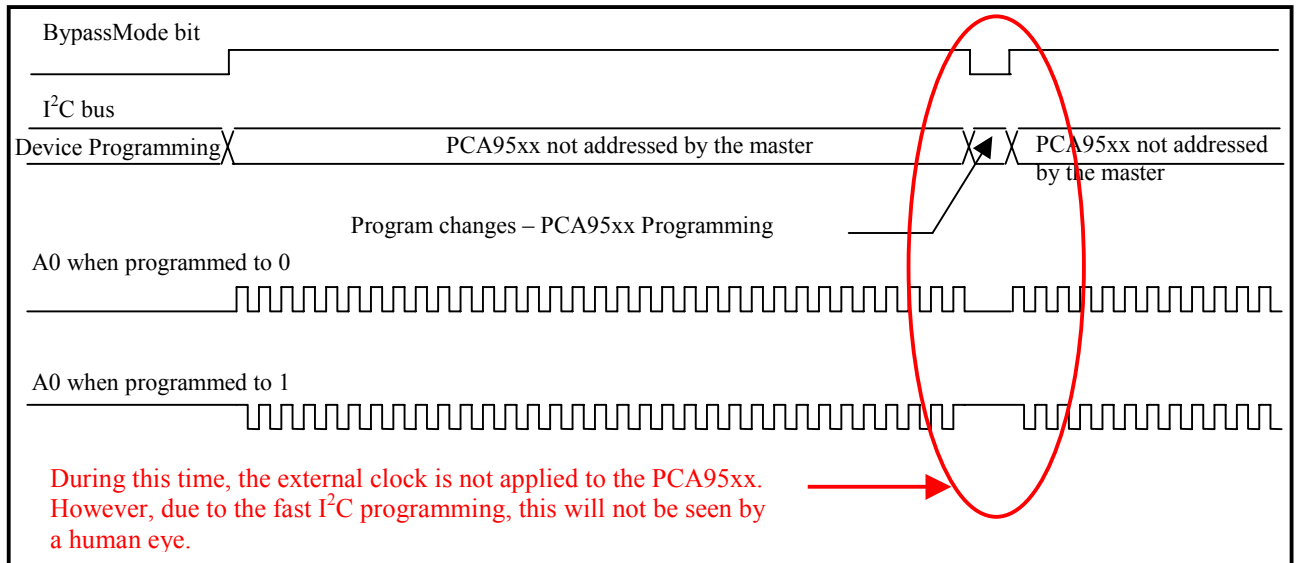


Figure 12. Bypass Mode waveforms

General Purpose I/O

The LED Dimmers and Blinkers feature open drain outputs that sink 25 mA per bit with a maximum of 50 mA for the PCA9530/50, 100 mA for the PCA9531/33/51/53 and 200 mA (100 mA per 8-bit group) for the PCA9532/52. Any bits

not used to control LEDs can be used as normal general purpose I/O bits. For more information, please refer to the paragraph “How to use unused bits as General Purpose I/O”.

PCA955x LED Blinker Typical Application

Figure 13 shows the PCA9551 in a typical LED blinking application.

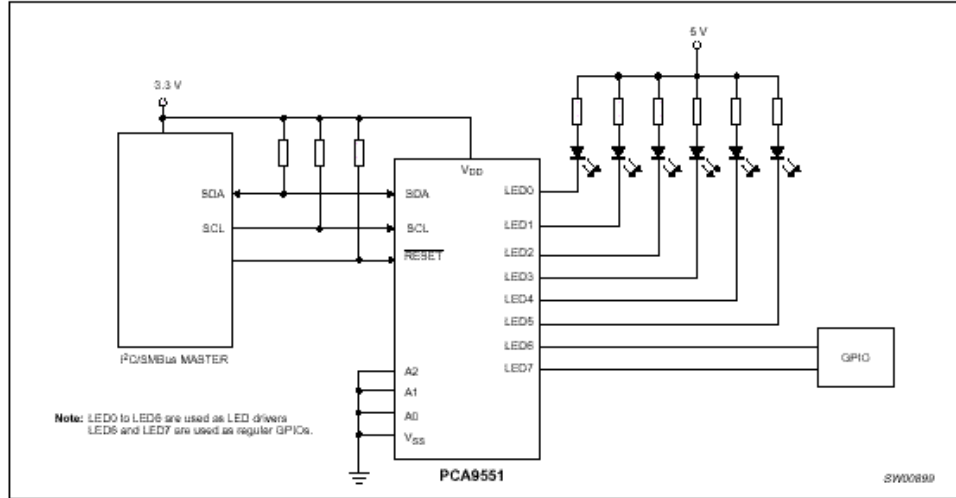


Figure 13. PCA9551 in a Typical LED blinking Application

PCA95xx LED Dimmer or Blinker using a charge pump and external driver

Figure 14 shows the PCA9551 in an application where a charge pump is needed to boost up the supply voltage for white and/or blue LEDs and/or an external device is needed to sink more current than the LED Blinker is capable of sinking. This type of external device assistance is more applicable when the LED Dimmers or Blinkers are operating at lower voltages since less current can be sunk by the device and the voltage drop across the white and blue LEDs is generally around four volts.

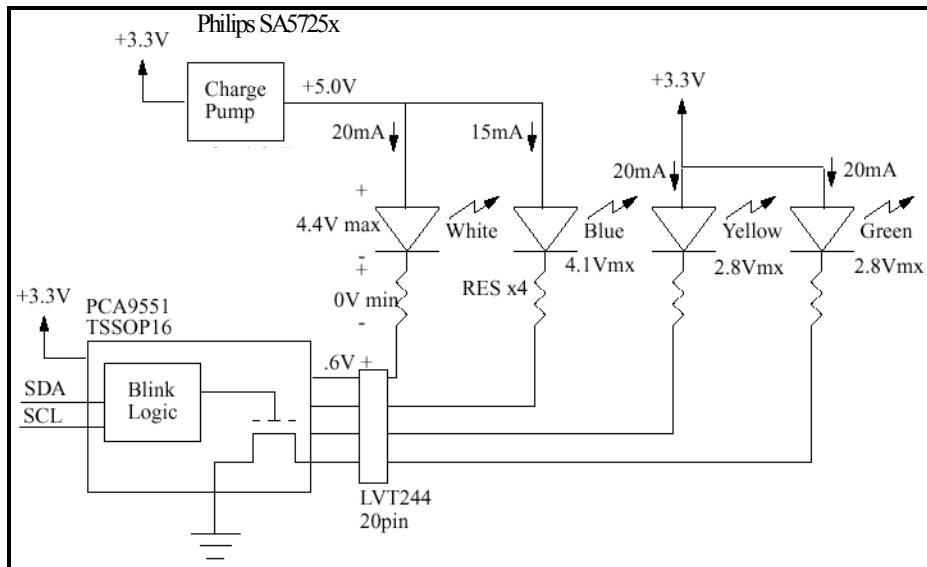


Figure 14. PCA9551 in application with charge pump and external source

SAA1064 Typical Application

Figure 15 shows the SAA1064 in a typical application to drive four 7-segment displays with decimal or 32 LEDs by multiplexing the supply through MX1 and MX2 control of the FET. Two 7-segment displays with decimal or 16 LEDs can be controlled statically.

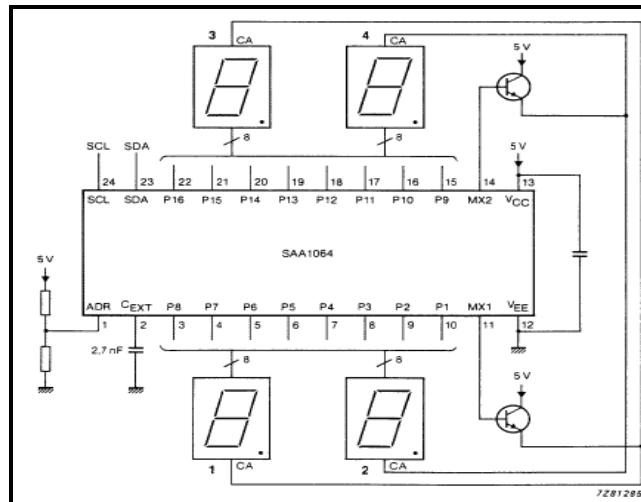


Figure 15. SAA1064 in a Typical Application

LED BLINKER EVALUATION BOARD

Purpose

The I²C 2002-1 evaluation board allows control and evaluation of our newest I²C devices including the PCA9550 and PCA9551 LED Blinkers. Controlled by PC-based software, the solution allows designers to test the different registers of the devices and develop applications or demonstrations.

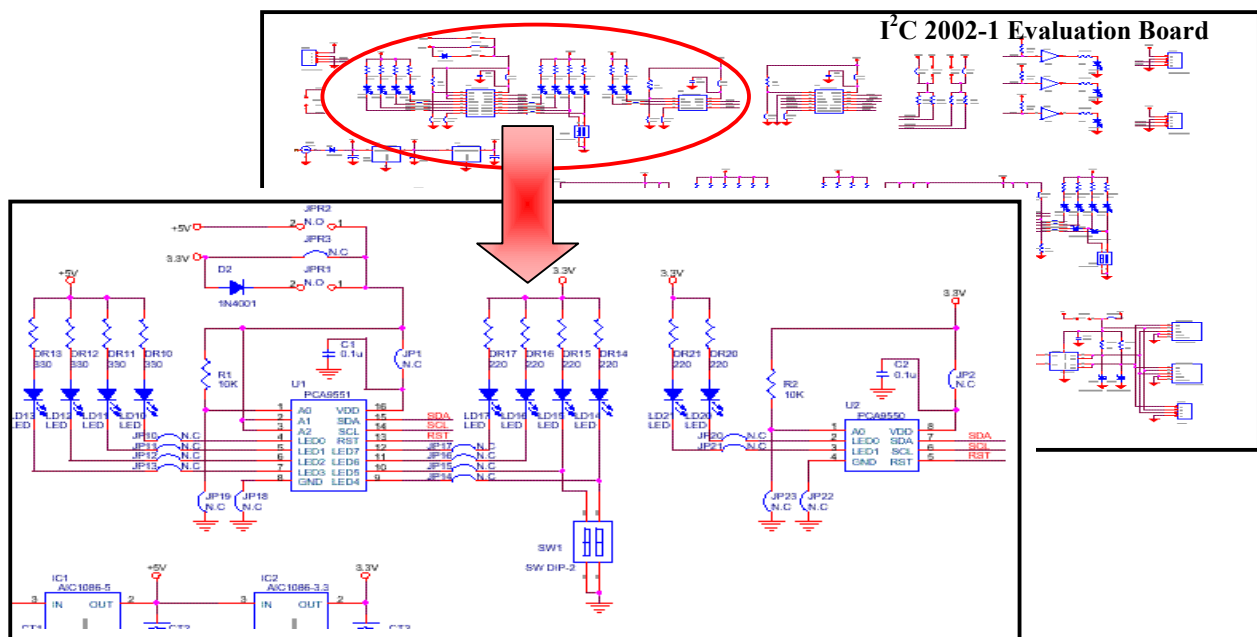


Figure 16. I²C 2002-1 Evaluation Board: Schematic LED Blinkers PCA9550 – PCA9551

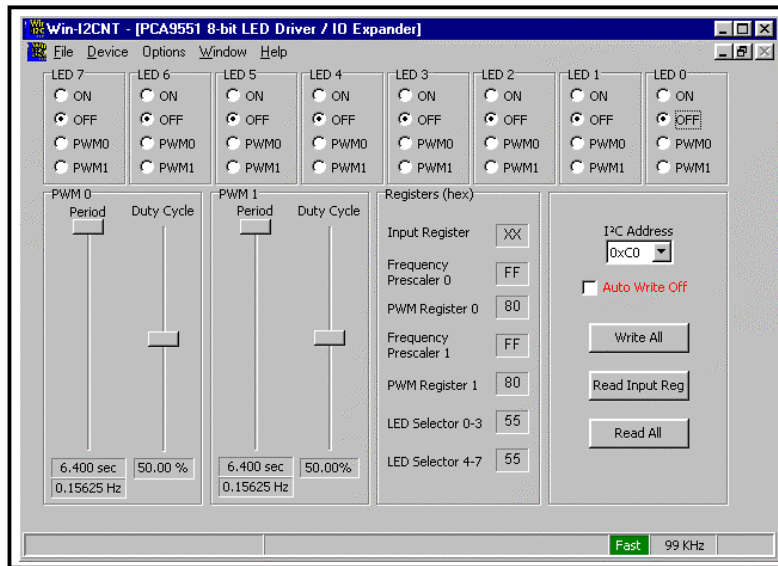


Figure 17. I²C 2002-1 Evaluation Board: Software controlling LED Blinkers

Evaluation Board Purchase

The I2C2002-1 Evaluation Board, which includes the PC-controller, software, evaluation board, power supply and cable can be purchased at The Boardshop (www.demoboard.com) and is detailed in application note AN10146-01 I2C2002-1 Evaluation Board.

FREQUENTLY ASKED QUESTIONS

- Question:** I need to test all the LEDs hooked up to the LED Blinker. How can I easily turn all the LEDs on at once?
Answer: Send the following code sequence: Address (0xEX) – Control Register (0x15) – Data (0x00) – Data (0x00).
 Start
 1100XXX0 PCA9552 Address + Write
 00010101 AI = 1, Reg#5
 00000000 LED0 to LED3 ON
 00000000 LED4 to LED7 ON
 Stop
- Question:** What types of applications would use the LED Dimmers and Blinkers?
Answer: The LED Blinkers were originally designed to blink LED status indications in server applications. The LED Dimmers can be used to dim LEDs for backlight or RGB color mixing. Both the Dimmers and Blinkers can be used for general purpose LED control. Some applications include servers, cell phones, telecom and networking infrastructures.
- Question:** Can the LED Dimmers or Blinkers be used to blink multiple LEDs?
Answer: Each bit is limited to 25 mA, the 8-bit device can provide up to 100 mA total and the 16-bit device can provide up to 200 mA total (100 mA per 8-bit group). You can put as many LEDs on each bit and package as not to exceed these limits. Normally there will be one LED on each bit and each LED can be ON, OFF or at Blink Rate 1 or 2. An external FET or a bipolar transistor controlled by the LED blinker output signal can be used to increase the number of LEDs per output or to pull up the LED supply voltage to some voltage level higher than 6.0 V.
- Question:** Can the LED Dimmers or Blinkers be programmed at only two rates?
Answer: Yes, each device can only have two programmed blink rates at one time but different devices can have

different blink rates and the same device can be reprogrammed on the fly to change the blink rates

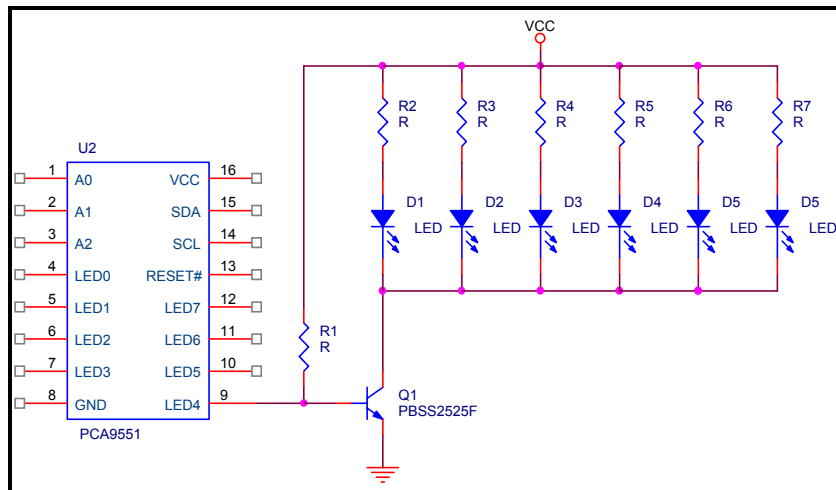
5. **Question:** We require lighting up 18 LEDs in 3 different colors in our cell phone. 6 LEDs for each color and all will light up together. Our initial plan was to tie 6 LEDs to 1 or 2 outputs of the PCA9551, but the drive current may not be sufficient. We could go to the PCA9552 for more current capacity but it would create more space issues even in the HVQFN package. What do you recommend?

Answer: Use an off-chip control for the LEDs as shown in the diagram below. Q1 is Philips BISS transistor (part number PBSS2525F). It can support up to 5.0 A. R2 ~ R7 are the current limitation resistors. R1 pulls up to V_{CC} , because LED0~7 are Open Drain outputs. If space is a big concern, use the PCA9553DP (4-bit LED blinker), one BISS transistor for each of the colors.

Note: the polarity is now inverted compared to the regular LED blinkers.

→ When LEDout = 0, the LEDs are OFF

→ When LEDout = 1, the LEDs are ON.



6. **Question:** Can you tell me if the PCA9551 or PCA9553 can be used to control a 2-wire bicolor LED (back to back configuration)?

Answer: The main problem with those diodes is that you need to be able to sink AND source current (back to back configuration for the 2 LEDs). Our output drivers have an open-drain structure, which means that they have only a sink capability so the circuit shown in figure 7 needs to be used.

7. **Question a:** I am using the SAA1064 to drive a bank of LEDs. The data sheet shows a Bipolar on the Mux pin, and says that the current is programmable through the diodes.

Is there a specific bipolar device that you recommend, or can we use an N-Channel MOSFET?

Answer a: A standard NPN transistor is recommended as an inexpensive solution for this application. An N-channel FET would not work very well. The choice of transistor depends upon the current you will be driving with the SAA1064. Assuming that each segment is receiving (for example) 18mA current and you are driving 16 segments at a time; you need a transistor rated for at least 288 mA peak current. Something like the BCP68 will work well.

Question b: How do you detect the current through each diode? Do we need to limit the current by putting a resistor in series with the diode?

Answer b: The SAA1064 uses a programmable current sink so no limiting resistors are required.

8. **Question:** We are using the SAA1064 to drive a series of LEDs, which we will be updating approximately 30 times per second. We also have PCA9555s, and PCA9554s on the bus, as well as devices from other vendors. Is there an I²C spec stating the longest time a slave can hold the SCL line low putting the master in a wait state? If an I²C device holds the bus for a longer than expected period of time, we may not be able to refresh the LED display at the refresh rate we are interested in. By refresh, I mean modifying the LED outputs on the SAA1064; not the self-refresh achieved by the MUX pins on the part.

Answer: There should be no problem refreshing the SAA1064 at 30 Hz, however, without knowing the exact part types on the bus, it is not known for certain. Philips Semiconductors parts will not cause a problem. The I²C spec does not specify a maximum low time for SCL.

ADDITIONAL INFORMATION

The latest datasheets for the PCA953x family of LED Dimmers, PCA955x family of LED Blinkers, SAA1064 and other SMBus/I²C products can be found at the Philips Semiconductors website:

<http://www.philipslogic.com/i2c>

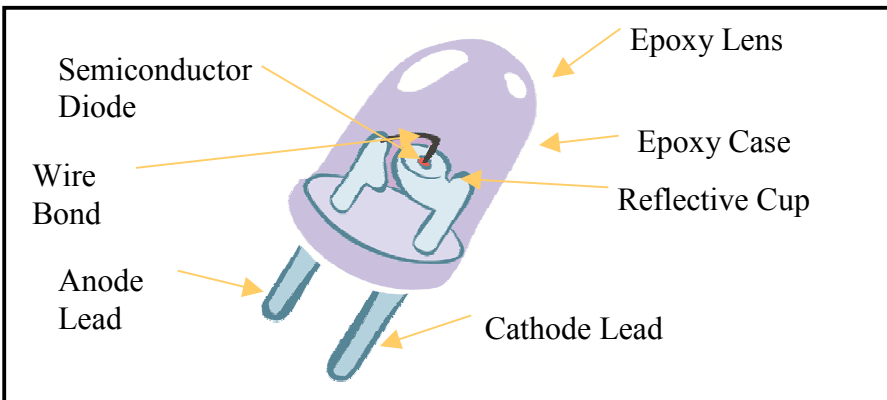
Software tools for most of Philips' products can be found at:

<http://www.demoboard.com>

Additional technical support for PCA953x LED Dimmers, PCA955x LED Blinkers and SAA1064 can be provided by e-mailing the question to:

Email: pc.mb.svl@philips.com

APPENDIX 1 - LED OVERVIEW



Interesting information related to the physics and technology can be found at the following web sites:

<http://www.lightemittingdiodes.org>

<http://www.howstuffworks.com/led.htm>

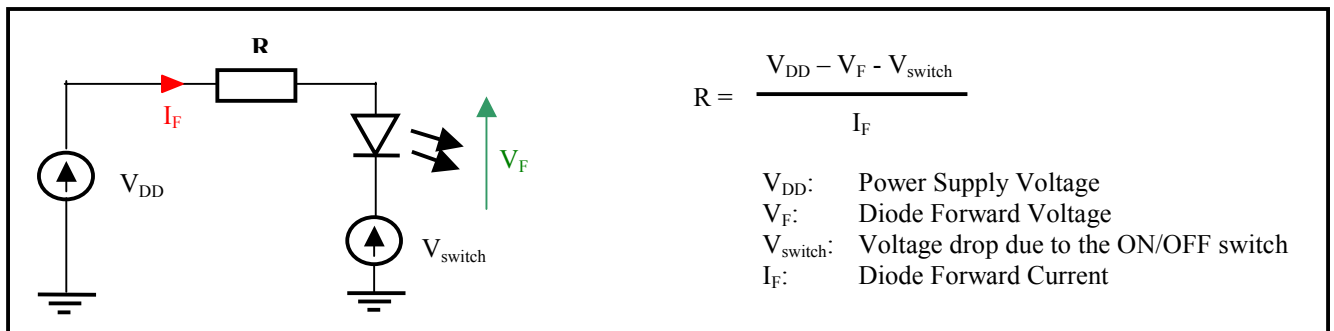
<http://www.lumileds.com>

APPENDIX 2 - LED RESISTOR CALCULATION

- Serial resistor Calculation

I_F determines the amount of current flowing through the LED that directly determines the brightness of the device.

V_{SWITCH} is the voltage drop across the LED Dimmer or Blinker I/O that is typically 0.7V.



- Forward voltage characteristics

LED Color	Forward Voltage (V_F) At $I_F = 20$ mA
Red	1.7 to 2.5 V
Orange	
Yellow	
Yellow Green	
Green	
Pure Green	
Blue	3.5 to 4.0 V
Blue Green	3.5 to 4.0 V
White	

APPENDIX 3 – RGB COLOR MIXING

Millions of colors and color changing effects can be generated from red, green and blue LEDs by additive color mixing. The amount of red, green or blue added to the mixed color is controlled by the brightness of the red, green, or blue LEDs.

LEDs controlled by the PCA953x LED Dimmer would have 256 levels of brightness since there are 256 steps in the duty cycle and the brightness is linearly proportionally to the average current through the LED. Three LEDs, one green, one blue, and one red (as shown by the corner of the triangle) would therefore yield over 16 million different colors ($256 \times 256 \times 256 = 16,777,216$).

