



Tech Support: services@elecfreaks.com

## Ultrasonic Ranging Module HC - SR04

### Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

### Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

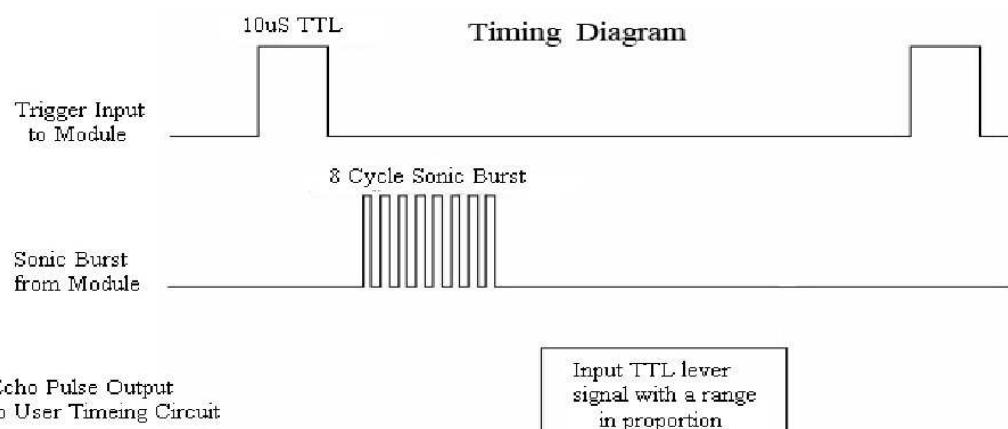
### Electric Parameter

<b>Working Voltage</b>	<b>DC 5 V</b>
<b>Working Current</b>	<b>15mA</b>
<b>Working Frequency</b>	<b>40Hz</b>
<b>Max Range</b>	<b>4m</b>
<b>Min Range</b>	<b>2cm</b>
<b>MeasuringAngle</b>	<b>15 degree</b>
<b>Trigger Input Signal</b>	<b>10uS TTL pulse</b>
<b>Echo Output Signal</b>	<b>Input TTL lever signal and the range in proportion</b>
<b>Dimension</b>	<b>45*20*15mm</b>



## Timing diagram

The Timing diagram is shown below. You only need to supply a short 10 $\mu$ s pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula:  $\mu\text{s} / 58 = \text{centimeters}$  or  $\mu\text{s} / 148 = \text{inch}$ ; or: the range = high level time \* velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



---

## **Attention:**

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

**[www.ElecFreaks.com](http://www.ElecFreaks.com)**

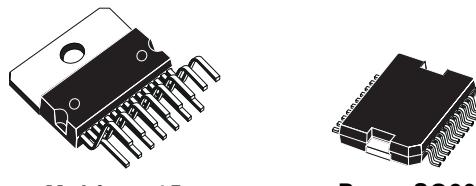


## DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V  
(HIGH NOISE IMMUNITY)

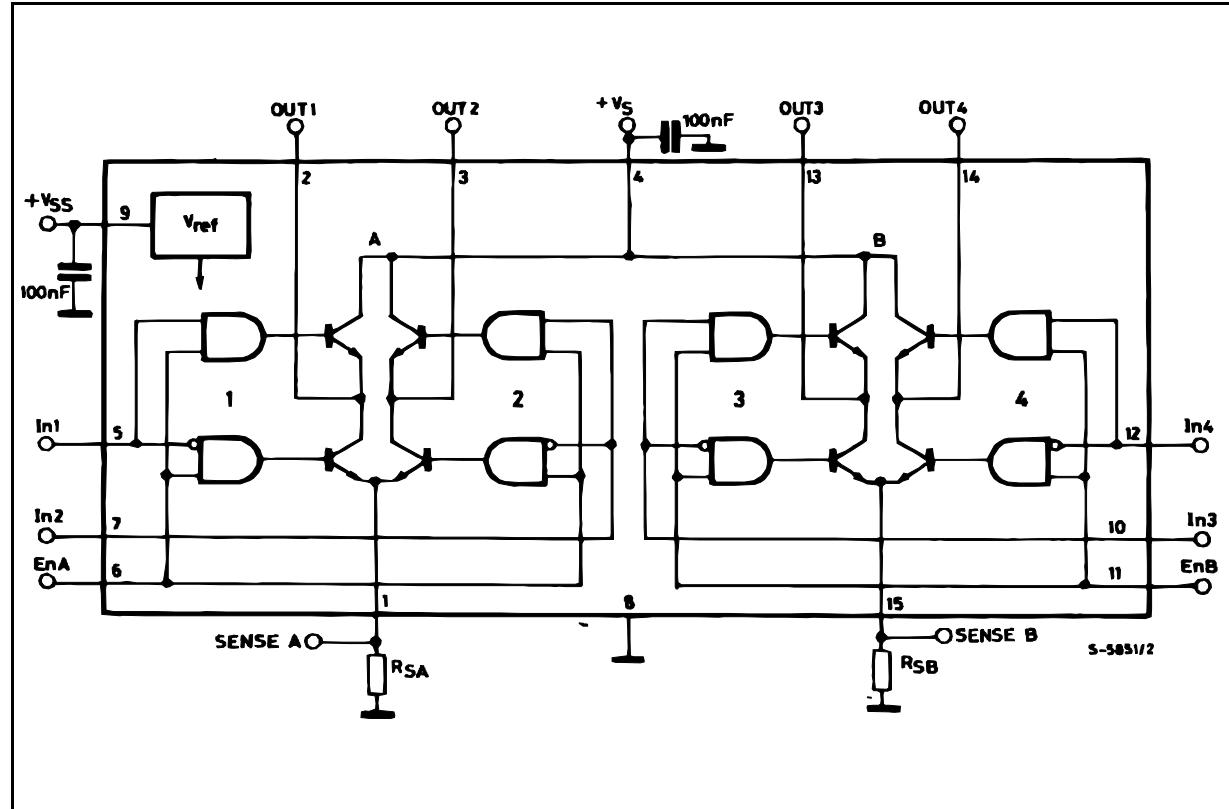
### DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.



**ORDERING NUMBERS :** L298N (Multiwatt Vert.)  
L298HN (Multiwatt Horiz.)  
L298P (PowerSO20)

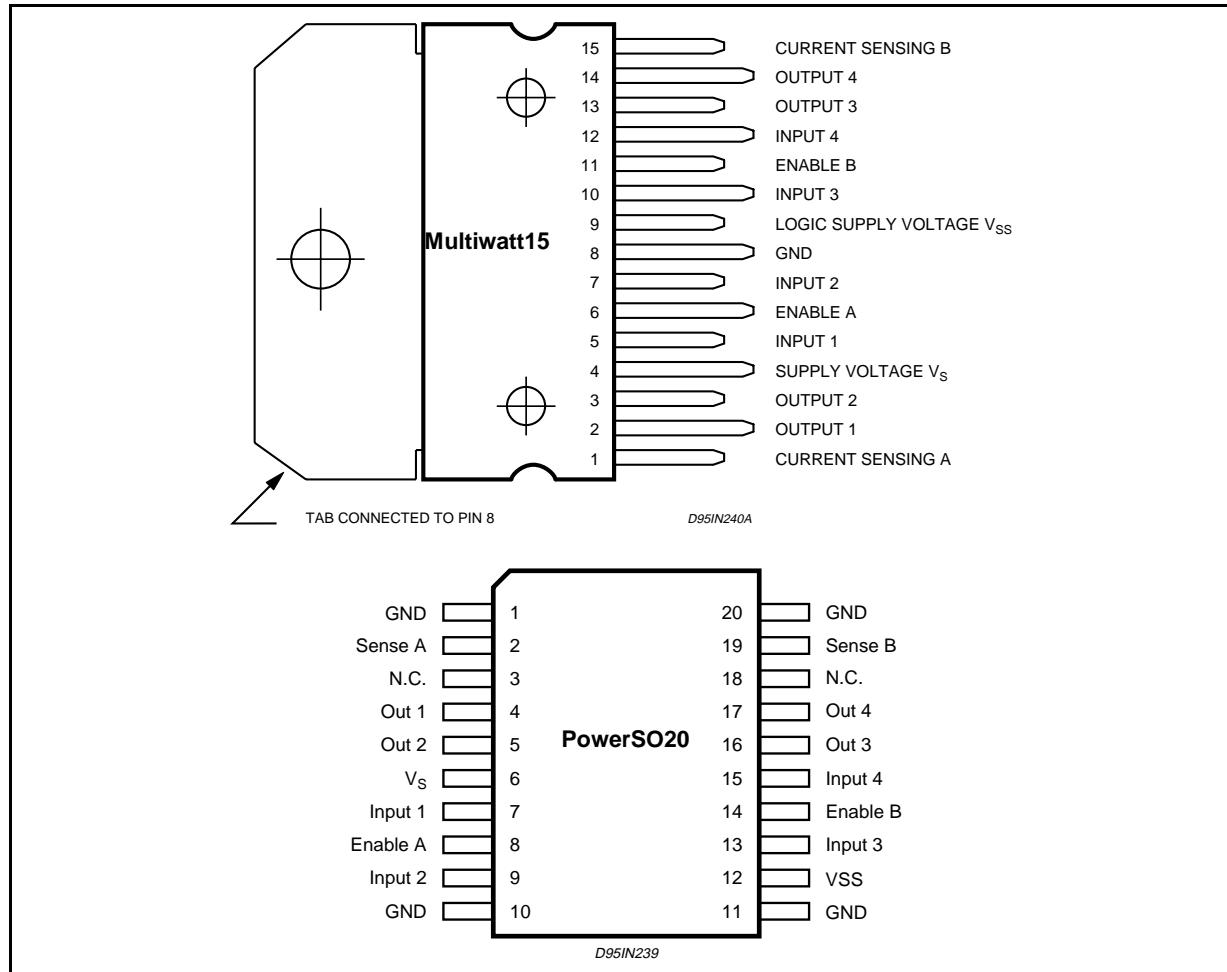
### BLOCK DIAGRAM



## ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
$V_s$	Power Supply	50	V
$V_{ss}$	Logic Supply Voltage	7	V
$V_i, V_{en}$	Input and Enable Voltage	-0.3 to 7	V
$I_o$	Peak Output Current (each Channel)		
	– Non Repetitive ( $t = 100\mu s$ )	3	A
	– Repetitive (80% on –20% off; $t_{on} = 10ms$ )	2.5	A
	– DC Operation	2	A
$V_{sens}$	Sensing Voltage	-1 to 2.3	V
$P_{tot}$	Total Power Dissipation ( $T_{case} = 75^\circ C$ )	25	W
$T_{op}$	Junction Operating Temperature	-25 to 130	°C
$T_{stg}, T_j$	Storage and Junction Temperature	-40 to 150	°C

## PIN CONNECTIONS (top view)



## THERMAL DATA

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
$R_{th j-case}$	Thermal Resistance Junction-case	Max.	–	3 °C/W
$R_{th j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35 °C/W

(\*) Mounted on aluminum substrate

**PIN FUNCTIONS** (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V <sub>s</sub>	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V <sub>SS</sub>	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

**ELECTRICAL CHARACTERISTICS** ( $V_S = 42V$ ;  $V_{SS} = 5V$ ,  $T_j = 25^\circ C$ ; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$V_S$	Supply Voltage (pin 4)	Operative Condition	$V_{IH} +2.5$		46	V
$V_{SS}$	Logic Supply Voltage (pin 9)		4.5	5	7	V
$I_S$	Quiescent Supply Current (pin 4)	$V_{en} = H; I_L = 0$ $V_i = L$ $V_i = H$		13 50	22 70	mA mA
		$V_{en} = L$ $V_i = X$			4	mA
$I_{ss}$	Quiescent Current from $V_{SS}$ (pin 9)	$V_{en} = H; I_L = 0$ $V_i = L$ $V_i = H$		24 7	36 12	mA mA
		$V_{en} = L$ $V_i = X$			6	mA
$V_{IL}$	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
$V_{IH}$	Input High Voltage (pins 5, 7, 10, 12)		2.3		$V_{SS}$	V
$I_{IL}$	Low Voltage Input Current (pins 5, 7, 10, 12)	$V_i = L$			-10	$\mu A$
$I_{IH}$	High Voltage Input Current (pins 5, 7, 10, 12)	$V_i = H \leq V_{SS} - 0.6V$		30	100	$\mu A$
$V_{en} = L$	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
$V_{en} = H$	Enable High Voltage (pins 6, 11)		2.3		$V_{SS}$	V
$I_{en} = L$	Low Voltage Enable Current (pins 6, 11)	$V_{en} = L$			-10	$\mu A$
$I_{en} = H$	High Voltage Enable Current (pins 6, 11)	$V_{en} = H \leq V_{SS} - 0.6V$		30	100	$\mu A$
$V_{CEsat(H)}$	Source Saturation Voltage	$I_L = 1A$ $I_L = 2A$	0.95	1.35 2	1.7 2.7	V V
$V_{CEsat(L)}$	Sink Saturation Voltage	$I_L = 1A (5)$ $I_L = 2A (5)$	0.85	1.2 1.7	1.6 2.3	V V
$V_{CEsat}$	Total Drop	$I_L = 1A (5)$ $I_L = 2A (5)$	1.80		3.2 4.9	V V
$V_{sens}$	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

**ELECTRICAL CHARACTERISTICS (continued)**

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T <sub>1</sub> (V <sub>i</sub> )	Source Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (2); (4)		1.5		μs
T <sub>2</sub> (V <sub>i</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		0.2		μs
T <sub>3</sub> (V <sub>i</sub> )	Source Current Turn-on Delay	0.5 V <sub>i</sub> to 0.1 I <sub>L</sub> (2); (4)		2		μs
T <sub>4</sub> (V <sub>i</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.7		μs
T <sub>5</sub> (V <sub>i</sub> )	Sink Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		0.7		μs
T <sub>6</sub> (V <sub>i</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.25		μs
T <sub>7</sub> (V <sub>i</sub> )	Sink Current Turn-on Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		1.6		μs
T <sub>8</sub> (V <sub>i</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.2		μs
f <sub>c</sub> (V <sub>i</sub> )	Commutation Frequency	I <sub>L</sub> = 2A		25	40	KHz
T <sub>1</sub> (V <sub>en</sub> )	Source Current Turn-off Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (2); (4)		3		μs
T <sub>2</sub> (V <sub>en</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		1		μs
T <sub>3</sub> (V <sub>en</sub> )	Source Current Turn-on Delay	0.5 V <sub>en</sub> to 0.1 I <sub>L</sub> (2); (4)		0.3		μs
T <sub>4</sub> (V <sub>en</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.4		μs
T <sub>5</sub> (V <sub>en</sub> )	Sink Current Turn-off Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (3); (4)		2.2		μs
T <sub>6</sub> (V <sub>en</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.35		μs
T <sub>7</sub> (V <sub>en</sub> )	Sink Current Turn-on Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (3); (4)		0.25		μs
T <sub>8</sub> (V <sub>en</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.1		μs

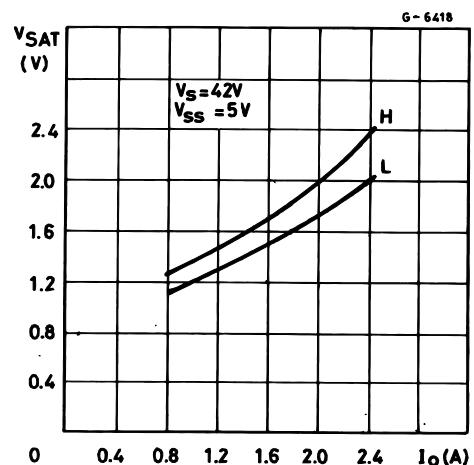
1) 1)Sensing voltage can be  $-1\text{ V}$  for  $t \leq 50\text{ μsec}$ ; in steady state  $V_{\text{sens}} \text{ min} \geq -0.5\text{ V}$ .

2) See fig. 2.

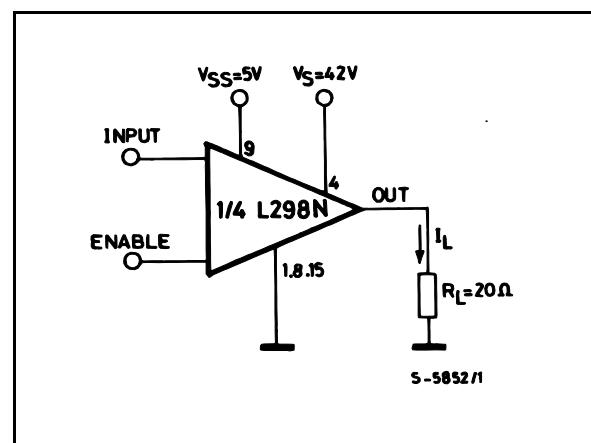
3) See fig. 4.

4) The load must be a pure resistor.

**Figure 1 : Typical Saturation Voltage vs. Output Current.**

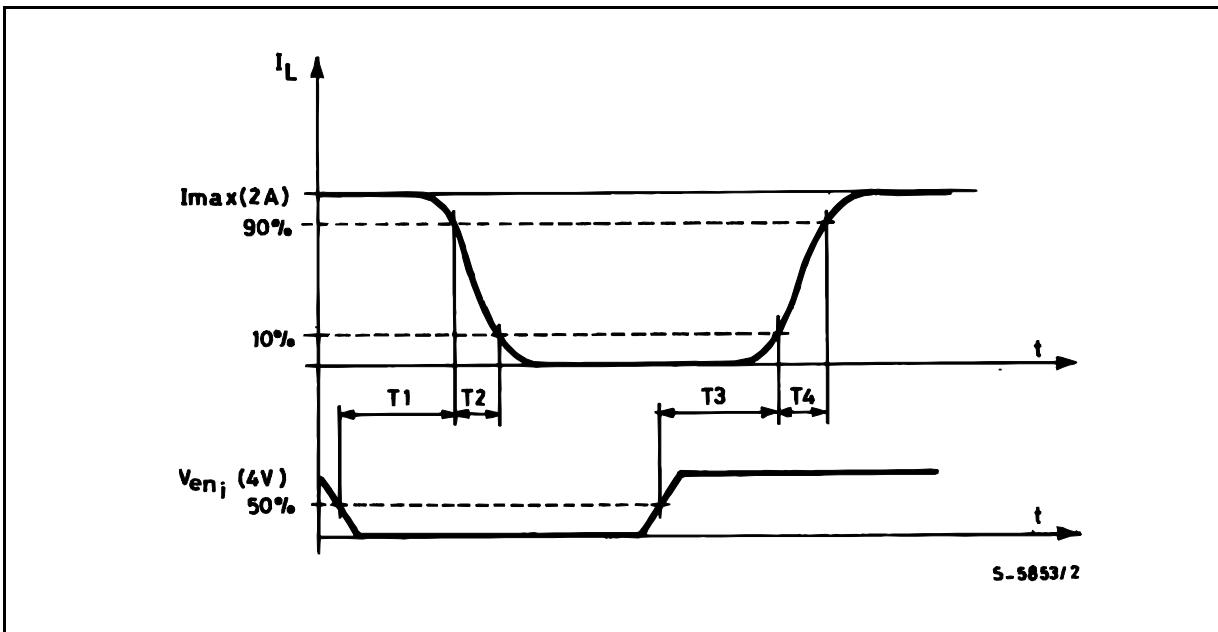


**Figure 2 : Switching Times Test Circuits.**

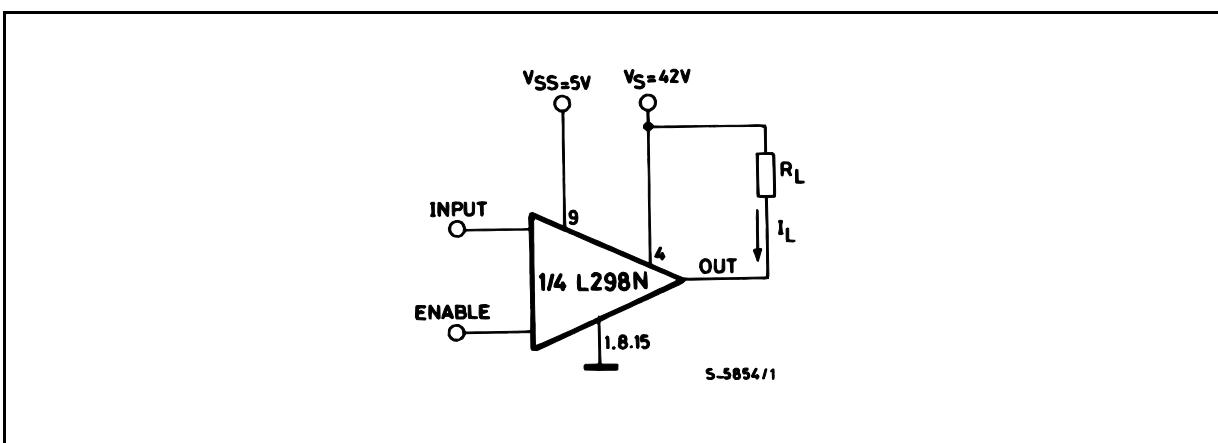


Note : For INPUT Switching, set EN = H  
For ENABLE Switching, set IN = H

**Figure 3 : Source Current Delay Times vs. Input or Enable Switching.**



**Figure 4 : Switching Times Test Circuits.**



Note : For INPUT Switching, set EN = H  
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

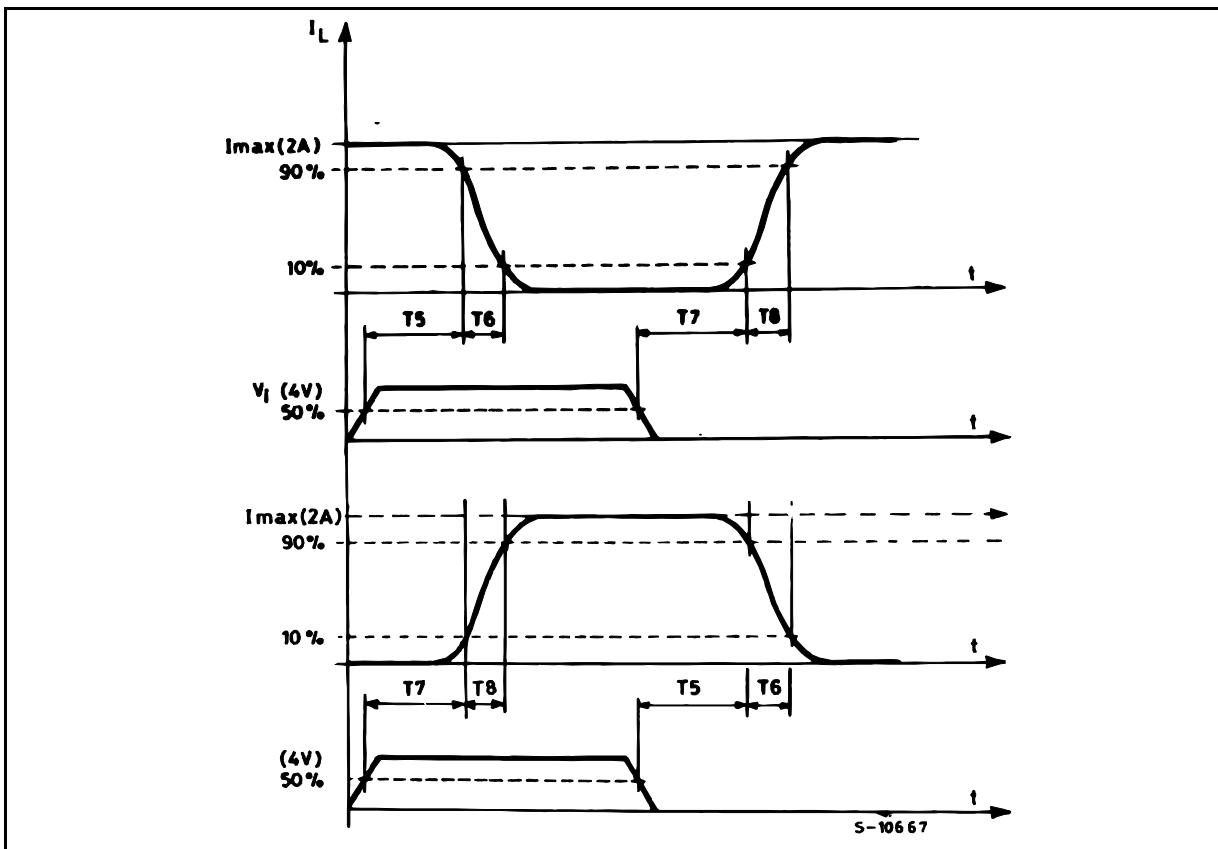
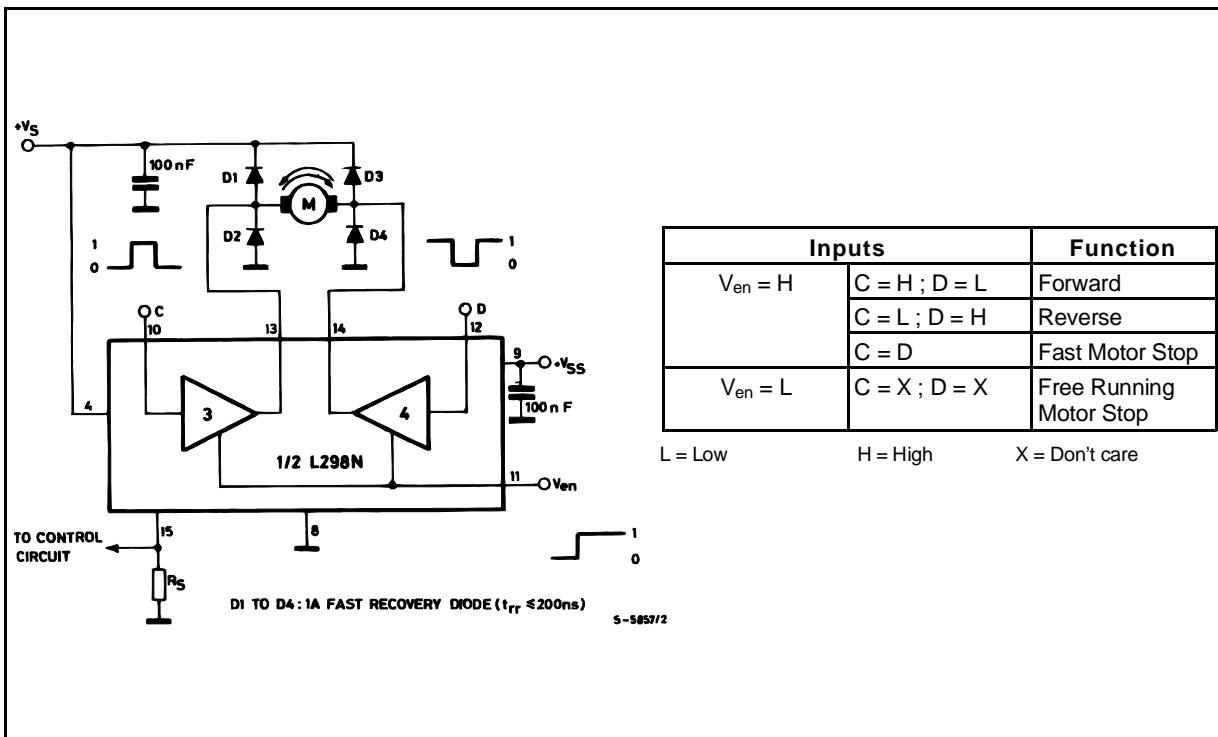
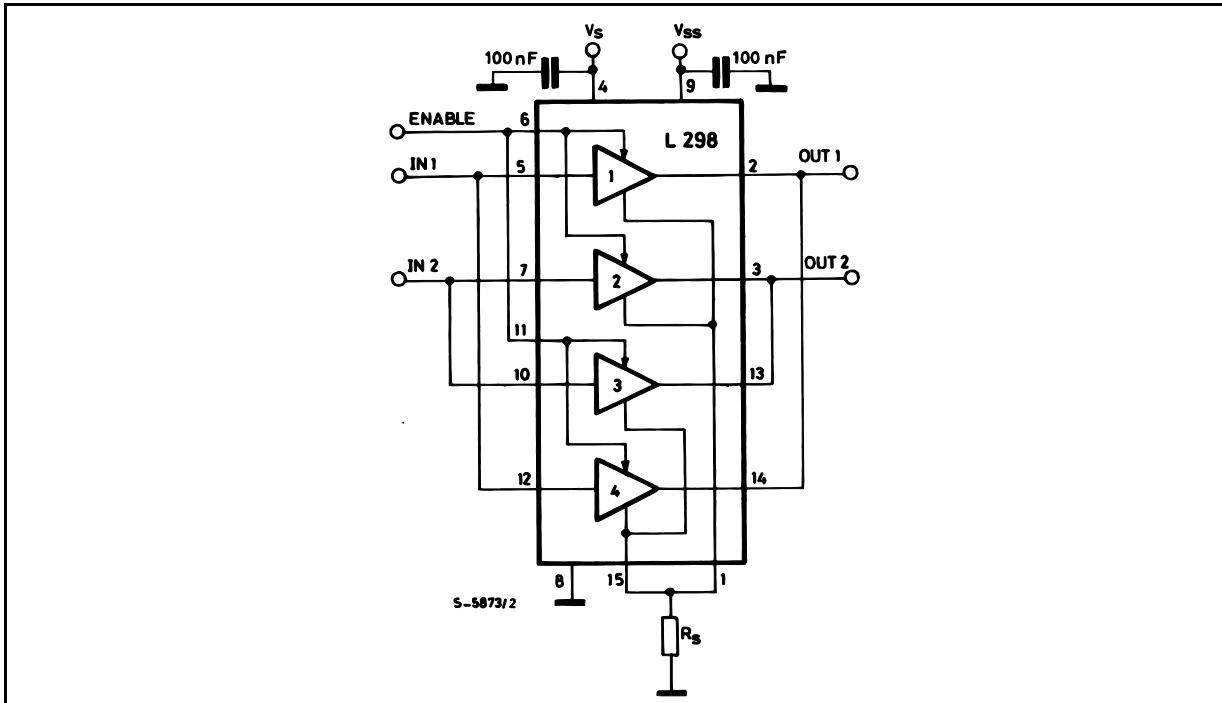


Figure 6 : Bidirectional DC Motor Control.



**Figure 7 :** For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



### APPLICATION INFORMATION (Refer to the block diagram)

#### 1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor ( $R_{SA}$  ;  $R_{SB}$ ) allows to detect the intensity of this current.

#### 1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are  $In_1$  ;  $In_2$  ;  $En_A$  and  $En_B$ . The  $In$  inputs set the bridge state when The  $En$  input is high ; a low state of the  $En$  input inhibits the bridge. All the inputs are TTL compatible.

#### 2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both  $V_S$  and  $V_{SS}$ , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of  $V_S$  that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

#### 3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ( $trr \leq 200$  nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Shottky diodes would be preferred.

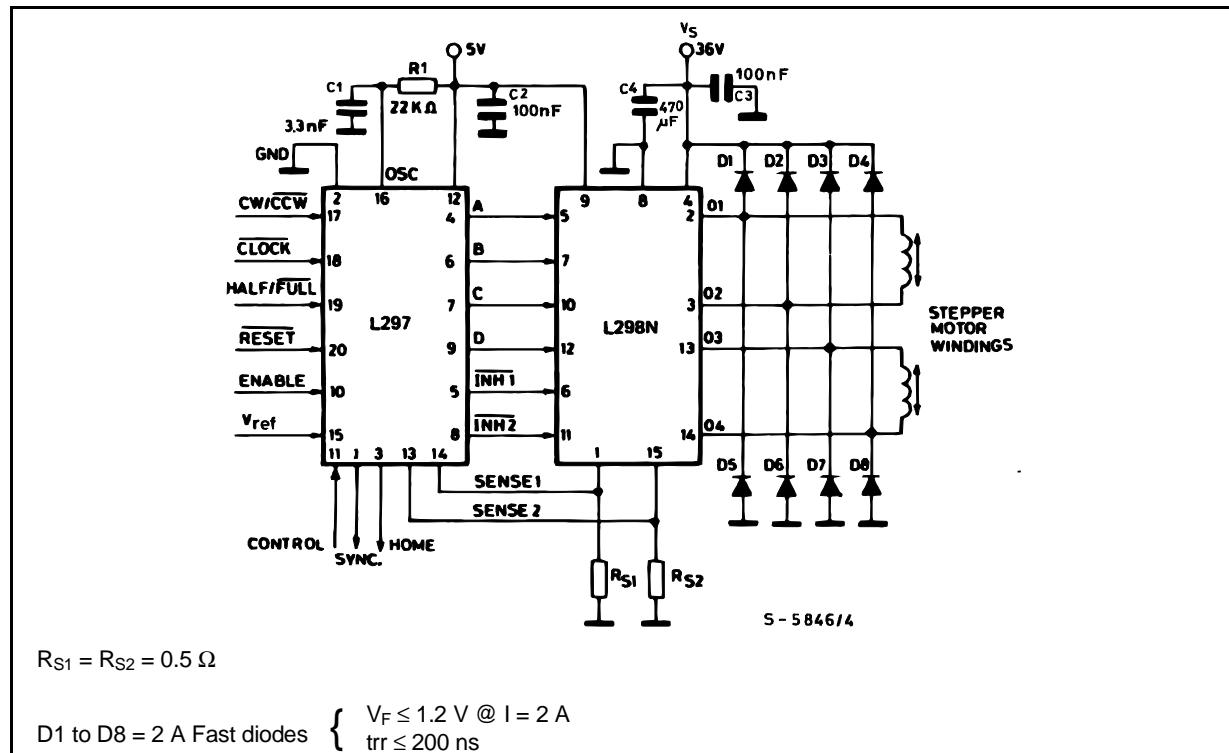
This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

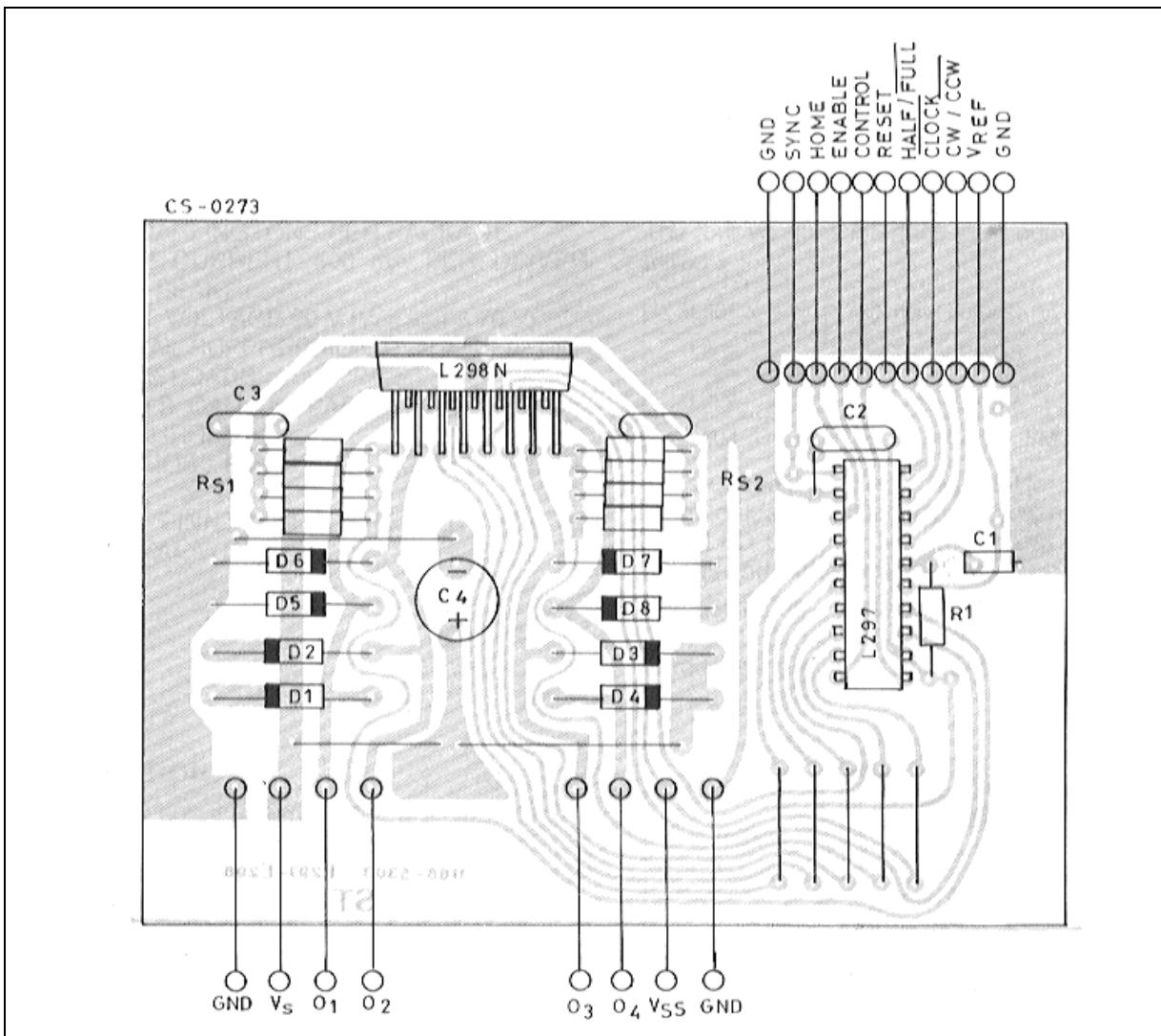
Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

**Figure 8 : Two Phase Bipolar Stepper Motor Circuit.**

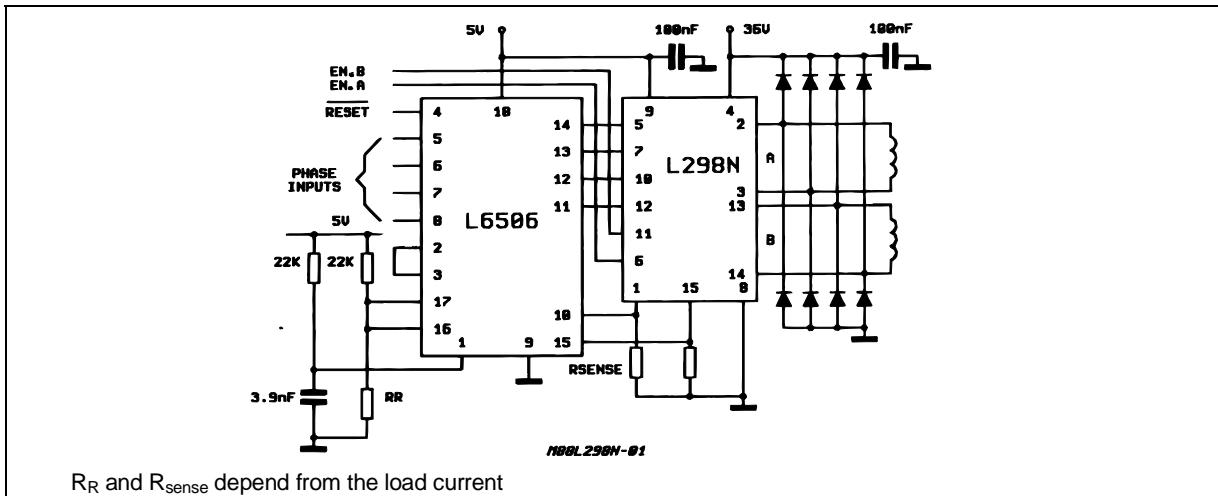
This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.



**Figure 9 :** Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

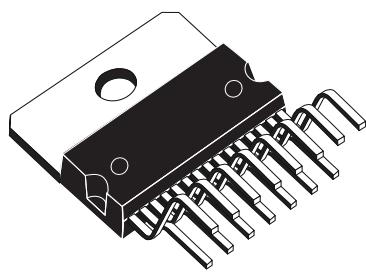


**Figure 10 :** Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.

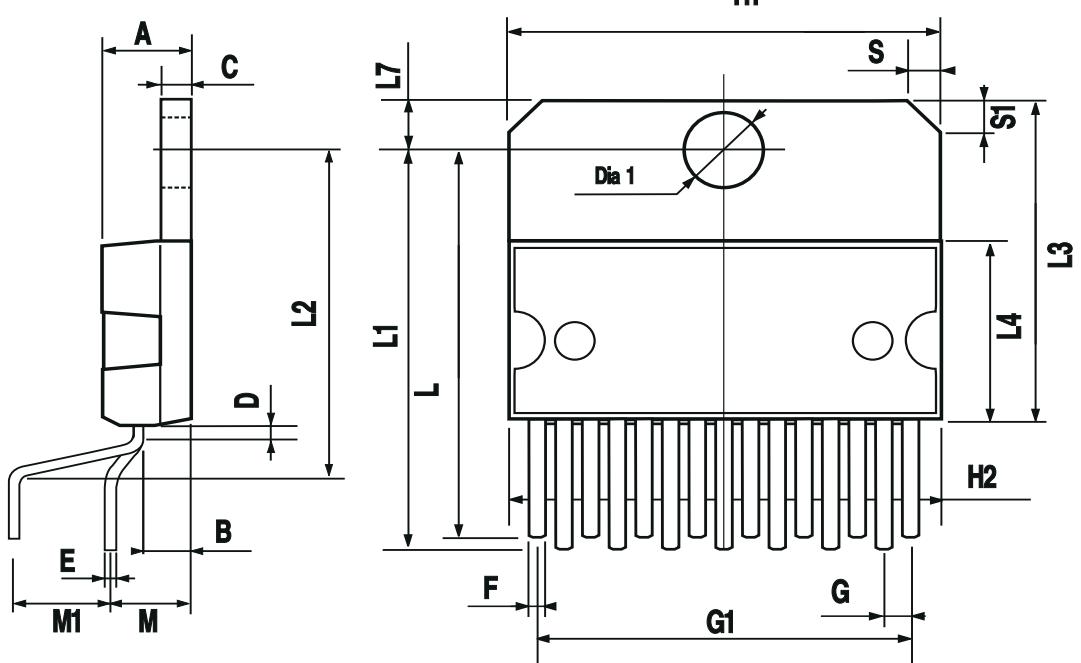


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

**OUTLINE AND  
MECHANICAL DATA**

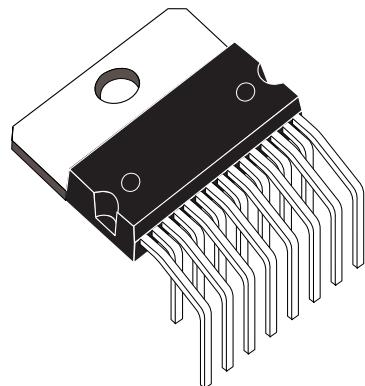


**Multiwatt15 V**

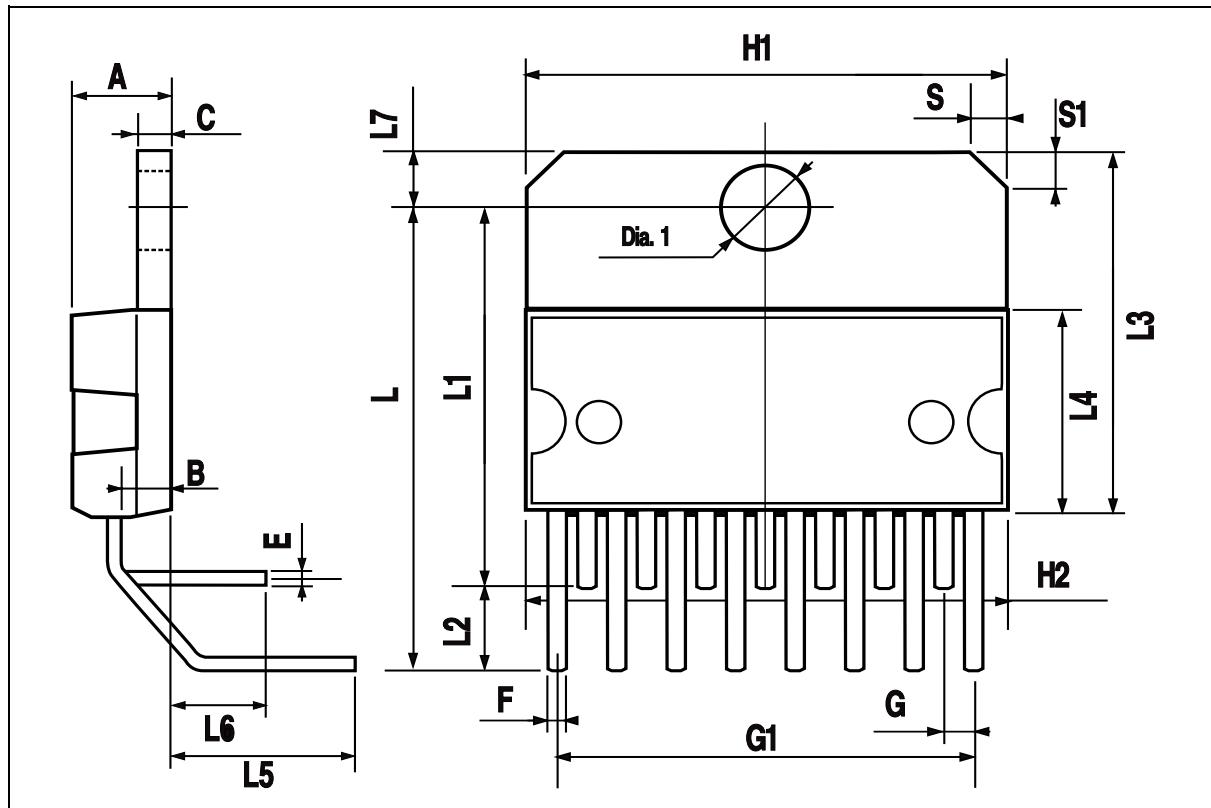


DIM.	mm			inch		
	MIN.	_TYP.	MAX.	MIN.	_TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

### OUTLINE AND MECHANICAL DATA



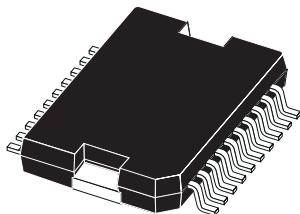
**Multiwatt15 H**



DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N			10° (max.)			
S			8° (max.)			
T		10			0.394	

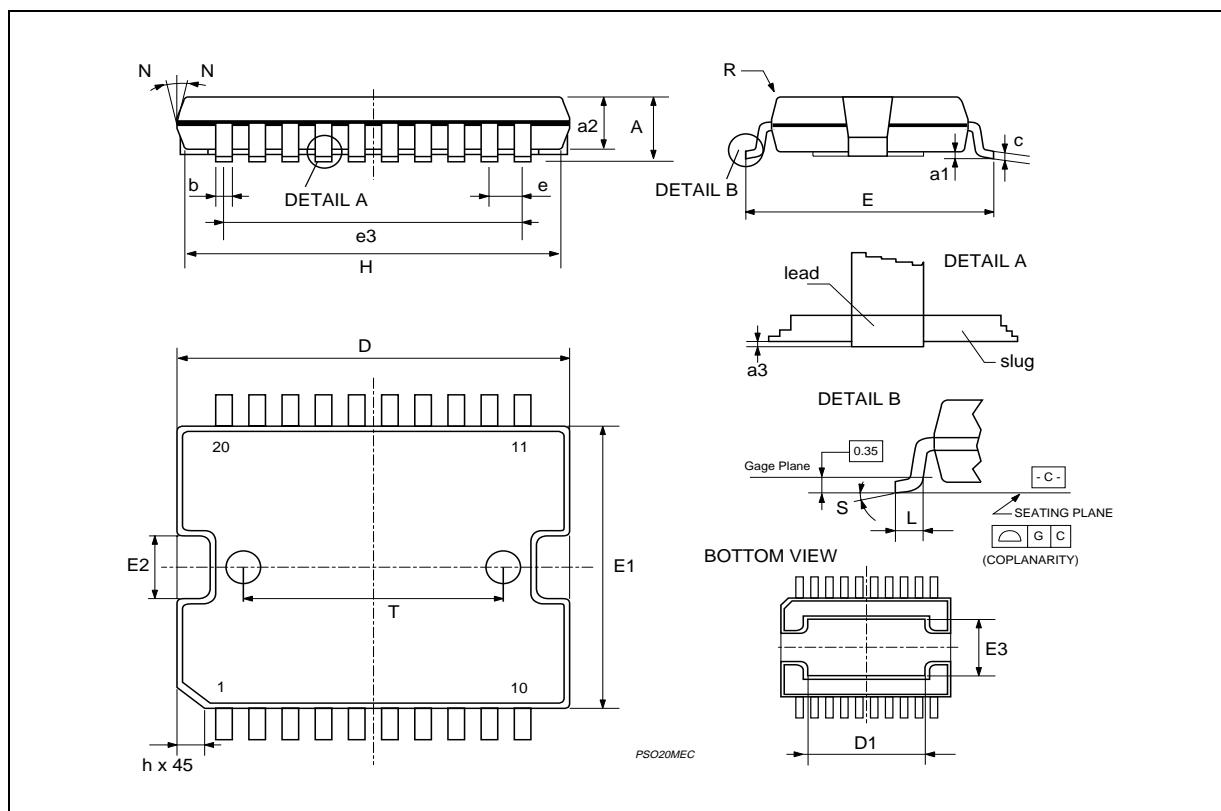
(1) "D and F" do not include mold flash or protrusions.  
- Mold flash or protrusions shall not exceed 0.15 mm (0.006").  
- Critical dimensions: "E", "G" and "a3"

## **OUTLINE AND MECHANICAL DATA**



JEDEC MO-166

# PowerSO20



Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics  
© 2000 STMicroelectronics – Printed in Italy – All Rights Reserved  
STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco -  
Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.  
<http://www.st.com>





MOTOROLA

Order this document by LM339/D

# Quad Single Supply Comparators

These comparators are designed for use in level detection, low-level sensing and memory applications in consumer automotive and industrial electronic applications.

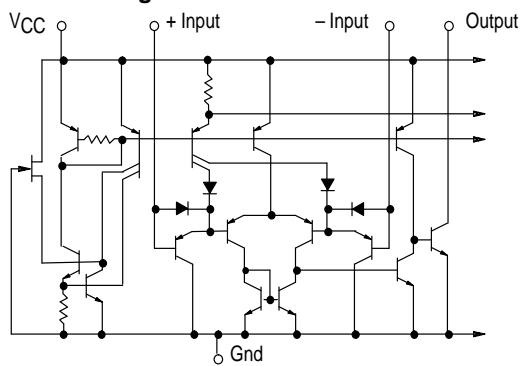
- Single or Split Supply Operation
- Low Input Bias Current: 25 nA (Typ)
- Low Input Offset Current:  $\pm 5.0$  nA (Typ)
- Low Input Offset Voltage:  $\pm 1.0$  mV (Typ) LM139A Series
- Input Common Mode Voltage Range to Gnd
- Low Output Saturation Voltage: 130 mV (Typ) @ 4.0 mA
- TTL and CMOS Compatible
- ESD Clamps on the Inputs Increase Reliability without Affecting Device Operation

## MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Power Supply Voltage LM239, A/LM339A/LM2901, V MC3302	V <sub>CC</sub>	+36 or $\pm 18$ +30 or $\pm 15$	Vdc
Input Differential Voltage Range LM239, A/LM339A/LM2901, V MC3302	V <sub>IDR</sub>	36 30	Vdc
Input Common Mode Voltage Range	V <sub>ICMR</sub>	-0.3 to V <sub>CC</sub>	Vdc
Output Short Circuit to Ground (Note 1)	I <sub>SC</sub>	Continuous	
Power Dissipation @ T <sub>A</sub> = 25°C Plastic Package Derate above 25°C	P <sub>D</sub>	1.0 8.0	W mW/°C
Junction Temperature	T <sub>J</sub>	150	°C
Operating Ambient Temperature Range LM239, A MC3302 LM2901 LM2901V LM339, A	T <sub>A</sub>	-25 to +85 -40 to +85 -40 to +105 -40 to +125 0 to +70	°C
Storage Temperature Range	T <sub>Stg</sub>	-65 to +150	°C

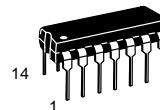
NOTE: 1. The maximum output current may be as high as 20 mA, independent of the magnitude of V<sub>CC</sub>. Output short circuits to V<sub>CC</sub> can cause excessive heating and eventual destruction.

Figure 1. Circuit Schematic

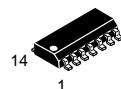


NOTE: Diagram shown is for 1 comparator.

**LM339, LM339A,  
LM239, LM239A,  
LM2901, M2901V,  
MC3302**

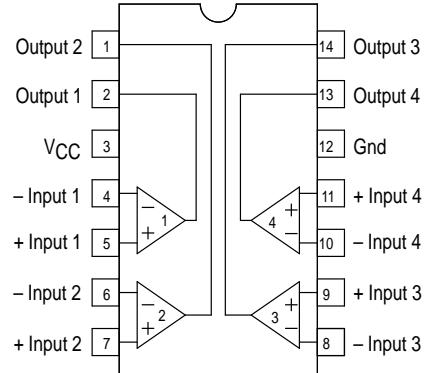


N, P SUFFIX  
PLASTIC PACKAGE  
CASE 646



D SUFFIX  
PLASTIC PACKAGE  
CASE 751A  
(SO-14)

## PIN CONNECTIONS



(Top View)

## ORDERING INFORMATION

Device	Operating Temperature Range	Package
LM239D,AD LM239N,AN	T <sub>A</sub> = 25° to +85°C	SO-14 Plastic DIP
LM339D, AD LM339N, AN	T <sub>A</sub> = 0° to +70°C	SO-14 Plastic DIP
LM2901D LM2901N	T <sub>A</sub> = -40° to +105°C	SO-14 Plastic DIP
LM2901VD LM2901VN	T <sub>A</sub> = -40° to +125°C	SO-14 Plastic DIP
MC3302P	T <sub>A</sub> = -40° to +85°C	Plastic DIP

# LM339, LM339A, LM239, LM239A, LM2901, M2901V, MC3302

**ELECTRICAL CHARACTERISTICS** ( $V_{CC} = +5.0$  Vdc,  $T_A = +25^\circ\text{C}$ , unless otherwise noted)

Characteristic	Symbol	LM239A/339A			LM239/339			LM2901/2901V			MC3302			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage (Note 4)	$V_{IO}$	—	$\pm 1.0$	$\pm 2.0$	—	$\pm 2.0$	$\pm 5.0$	—	$\pm 2.0$	$\pm 7.0$	—	$\pm 3.0$	$\pm 20$	mVdc
Input Bias Current (Notes 4, 5) (Output in Analog Range)	$I_{IB}$	—	25	250	—	25	250	—	25	250	—	25	500	nA
Input Offset Current (Note 4)	$I_{IO}$	—	$\pm 5.0$	$\pm 50$	—	$\pm 5.0$	$\pm 50$	—	$\pm 5.0$	$\pm 50$	—	$\pm 3.0$	$\pm 100$	nA
Input Common Mode Voltage Range	$V_{ICMR}$	0	—	$V_{CC} - 1.5$	0	—	$V_{CC} - 1.5$	0	—	$V_{CC} - 1.5$	0	—	$V_{CC} - 1.5$	V
Supply Current $R_L = \infty$ (For All Comparators) $R_L = \infty$ , $V_{CC} = 30$ Vdc	$I_{CC}$	—	0.8	2.0	—	0.8	2.0	—	0.8	2.0	—	0.8	2.0	mA
Supply Current $R_L = \infty$ (For All Comparators) $R_L = \infty$ , $V_{CC} = 30$ Vdc	$I_{CC}$	—	1.0	2.5	—	1.0	2.5	—	1.0	2.5	—	1.0	2.5	mA
Voltage Gain $R_L \geq 15$ k $\Omega$ , $V_{CC} = 15$ Vdc	$A_{VOL}$	50	200	—	50	200	—	25	100	—	25	100	—	V/mV
Large Signal Response Time $V_I$ = TTL Logic Swing, $V_{ref} = 1.4$ Vdc, $V_{RL} = 5.0$ Vdc, $R_L = 5.1$ k $\Omega$	—	—	300	—	—	300	—	—	300	—	—	300	—	ns
Response Time (Note 6) $V_{RL} = 5.0$ Vdc, $R_L = 5.1$ k $\Omega$	—	—	1.3	—	—	1.3	—	—	1.3	—	—	1.3	—	$\mu\text{s}$
Output Sink Current $V_I(-) \geq +1.0$ Vdc, $V_I(+) = 0$ , $V_O \leq 1.5$ Vdc	$I_{Sink}$	6.0	16	—	6.0	16	—	6.0	16	—	6.0	16	—	mA
Saturation Voltage $V_I(-) \geq +1.0$ Vdc, $V_I(+) = 0$ , $I_{sink} \leq 4.0$ mA	$V_{sat}$	—	130	400	—	130	400	—	130	400	—	130	500	mV
Output Leakage Current $V_I(+) \geq +1.0$ Vdc, $V_I(-) = 0$ , $V_O = +5.0$ Vdc	$I_{OL}$	—	0.1	—	—	0.1	—	—	0.1	—	—	0.1	—	nA

**PERFORMANCE CHARACTERISTICS** ( $V_{CC} = +5.0$  Vdc,  $T_A = T_{low}$  to  $T_{high}$  [Note 3])

Characteristic	Symbol	LM239A/339A			LM239/339			LM2901/2901V			MC3302			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage (Note 4)	$V_{IO}$	—	—	$\pm 4.0$	—	—	$\pm 9.0$	—	—	$\pm 15$	—	—	$\pm 40$	mVdc
Input Bias Current (Notes 4, 5) (Output in Analog Range)	$I_{IB}$	—	—	400	—	—	400	—	—	500	—	—	1000	nA
Input Offset Current (Note 4)	$I_{IO}$	—	—	$\pm 150$	—	—	$\pm 150$	—	—	$\pm 200$	—	—	$\pm 300$	nA
Input Common Mode Voltage Range	$V_{ICMR}$	0	—	$V_{CC} - 2.0$	0	—	$V_{CC} - 2.0$	0	—	$V_{CC} - 2.0$	0	—	$V_{CC} - 2.0$	V
Saturation Voltage $V_I(-) \geq +1.0$ Vdc, $V_I(+) = 0$ , $I_{sink} \leq 4.0$ mA	$V_{sat}$	—	—	700	—	—	700	—	—	700	—	—	700	mV
Output Leakage Current $V_I(+) \geq +1.0$ Vdc, $V_I(-) = 0$ , $V_O = 30$ Vdc	$I_{OL}$	—	—	1.0	—	—	1.0	—	—	1.0	—	—	1.0	$\mu\text{A}$
Differential Input Voltage All $V_I \geq 0$ Vdc	$V_{ID}$	—	—	$V_{CC}$	—	—	$V_{CC}$	—	—	$V_{CC}$	—	—	$V_{CC}$	Vdc

**NOTES:** 3. (LM239/239A)  $T_{low} = -25^\circ\text{C}$ ,  $T_{high} = +85^\circ\text{C}$

(LM339/339A)  $T_{low} = 0^\circ\text{C}$ ,  $T_{high} = +70^\circ\text{C}$

(MC3302)  $T_{low} = -40^\circ\text{C}$ ,  $T_{high} = +85^\circ\text{C}$

(LM2901)  $T_{low} = -40^\circ\text{C}$ ,  $T_{high} = +105^\circ\text{C}$

(LM2901V)  $T_{low} = -40^\circ\text{C}$ ,  $T_{high} = +125^\circ\text{C}$

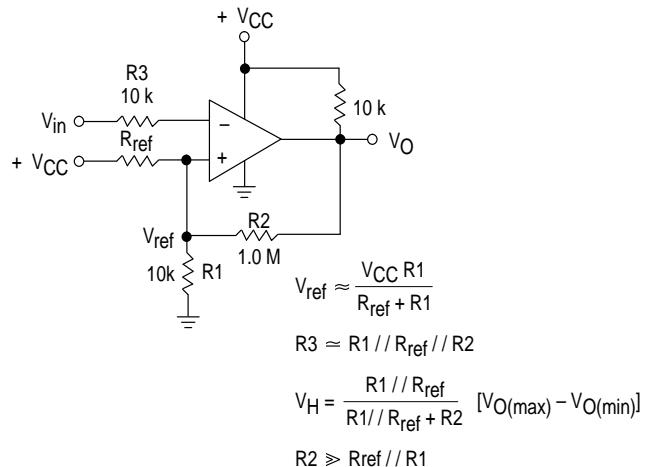
4. At the output switch point,  $V_O = 1.4$  Vdc,  $R_S \leq 100 \Omega$   $5.0$  Vdc  $\leq V_{CC} \leq 30$  Vdc, with the inputs over the full common mode range (0 Vdc to  $V_{CC} - 1.5$  Vdc).

5. The bias current flows out of the inputs due to the PNP input stage. This current is virtually constant, independent of the output state.

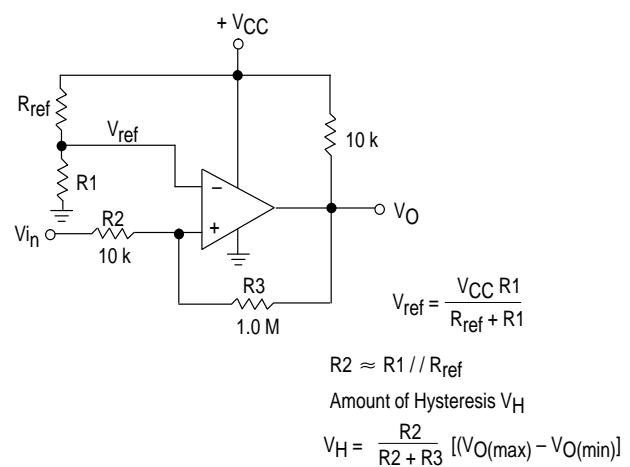
6. The response time specified is for a 100 mV input step with 5.0 mV overdrive. For larger signals, 300 ns is typical.

# LM339, LM339A, LM239, LM239A, LM2901, M2901V, MC3302

**Figure 2. Inverting Comparator with Hysteresis**



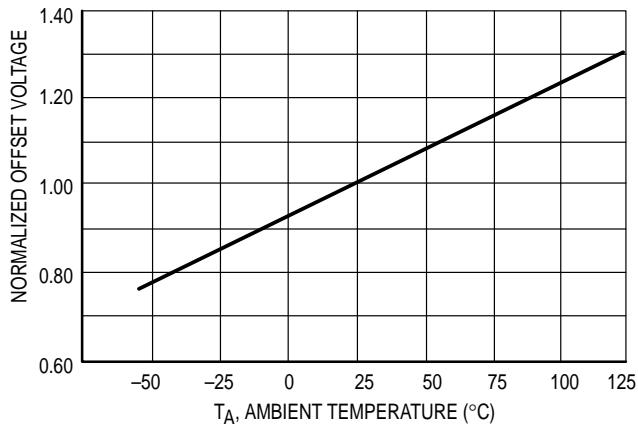
**Figure 3. Noninverting Comparator with Hysteresis**



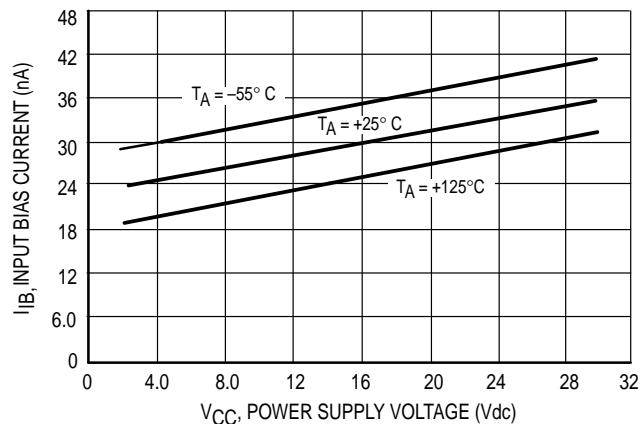
## Typical Characteristics

( $V_{CC} = 15$  Vdc,  $T_A = +25^\circ\text{C}$  (each comparator) unless otherwise noted.)

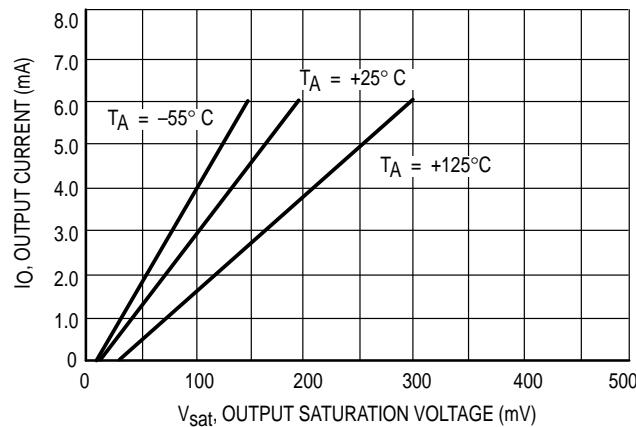
**Figure 4. Normalized Input Offset Voltage**



**Figure 5. Input Bias Current**

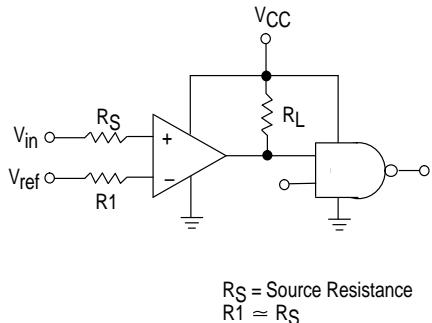


**Figure 6. Output Sink Current versus Output Saturation Voltage**



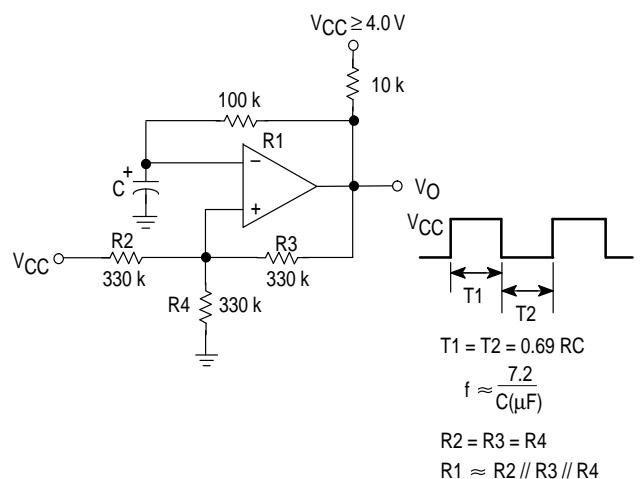
# LM339, LM339A, LM239, LM239A, LM2901, M2901V, MC3302

**Figure 7. Driving Logic**



Logic	Device	$V_{CC}$ (V)	$R_L$ k $\Omega$
CMOS	1/4 MC14001	+15	100
TTL	1/4 MC7400	+5.0	10

**Figure 8. Squarewave Oscillator**



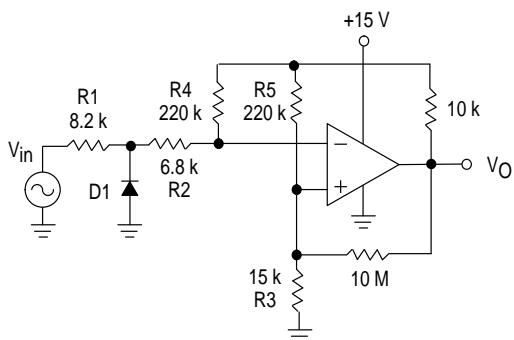
## APPLICATIONS INFORMATION

These quad comparators feature high gain, wide bandwidth characteristics. This gives the device oscillation tendencies if the outputs are capacitively coupled to the inputs via stray capacitance. This oscillation manifests itself during output transitions ( $V_{OL}$  to  $V_{OH}$ ). To alleviate this situation input resistors  $< 10 \text{ k}\Omega$  should be used. The addition

of positive feedback ( $< 10 \text{ mV}$ ) is also recommended. It is good design practice to ground all unused input pins.

Differential input voltages may be larger than supply voltages without damaging the comparator's inputs. Voltages more negative than  $-300 \text{ mV}$  should not be used.

**Figure 9. Zero Crossing Detector  
(Single Supply)**



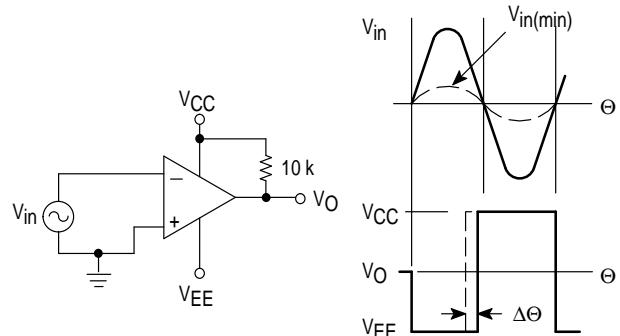
D1 prevents input from going negative by more than 0.6 V.

$$R_1 + R_2 = R_3$$

$$R_3 \leq \frac{R_5}{10} \text{ for small error in zero crossing}$$

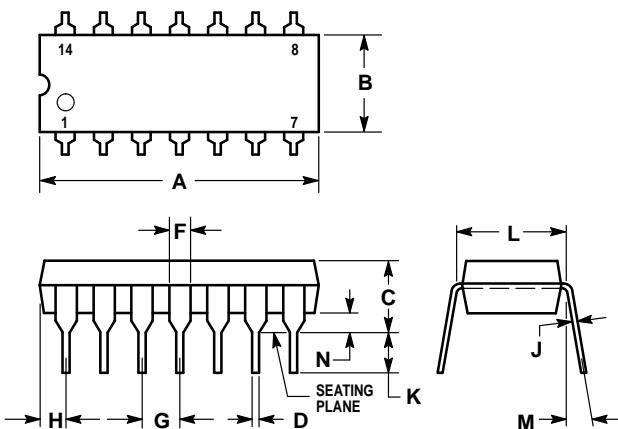
**Figure 10. Zero Crossing Detector  
(Split Supplies)**

$V_{in(min)} \approx 0.4 \text{ V peak for } 1\% \text{ phase distortion } (\Delta\Theta).$



**OUTLINE DIMENSIONS**

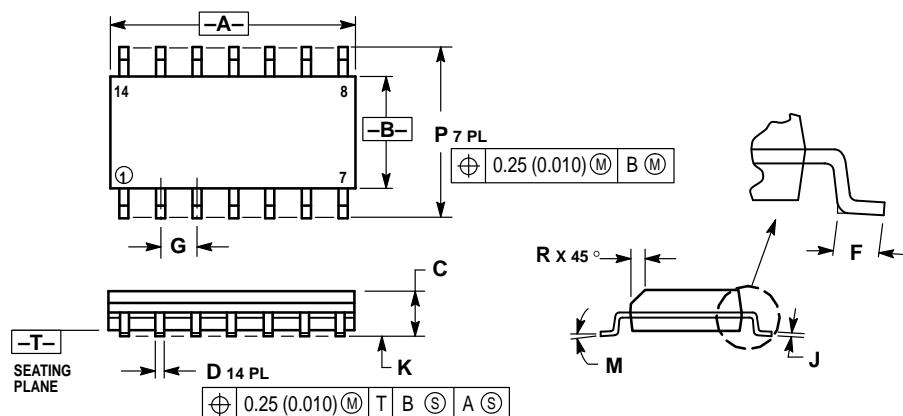
**N, P SUFFIX**  
PLASTIC PACKAGE  
CASE 646-06  
ISSUE L



- NOTES:
1. LEADS WITHIN 0.13 (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION.
  2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
  3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.
  4. ROUNDED CORNERS OPTIONAL.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.715	0.770	18.16	19.56
B	0.240	0.260	6.10	6.60
C	0.145	0.185	3.69	4.69
D	0.015	0.021	0.38	0.53
F	0.040	0.070	1.02	1.78
G	0.100 BSC		2.54 BSC	
H	0.052	0.095	1.32	2.41
J	0.008	0.015	0.20	0.38
K	0.115	0.135	2.92	3.43
L	0.300 BSC		7.62 BSC	
M	0°	10°	0°	10°
N	0.015	0.039	0.39	1.01

**D SUFFIX**  
PLASTIC PACKAGE  
CASE 751A-03  
(SO-14)  
ISSUE F



- NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
  2. CONTROLLING DIMENSION: MILLIMETER.
  3. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION.
  4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
  5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.127 (0.005) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	8.55	8.75	0.337	0.344
B	3.80	4.00	0.150	0.157
C	1.35	1.75	0.054	0.068
D	0.35	0.49	0.014	0.019
F	0.40	1.25	0.016	0.049
G	1.27 BSC		0.050 BSC	
J	0.19	0.25	0.008	0.009
K	0.10	0.25	0.004	0.009
M	0°	7°	0°	7°
P	5.80	6.20	0.228	0.244
R	0.25	0.50	0.010	0.019

# LM339, LM339A, LM239, LM239A, LM2901, M2901V, MC3302

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and  are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

**How to reach us:**

**USA/EUROPE/Locations Not Listed:** Motorola Literature Distribution;  
P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447 or 602-303-5454

**MFAX:** RMFAX0@email.sps.mot.com – **TOUCHTONE** 602-244-6609  
**INTERNET:** <http://Design-NET.com>

**JAPAN:** Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, 6F Seibu-Butsuryu-Center,  
3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-81-3521-8315

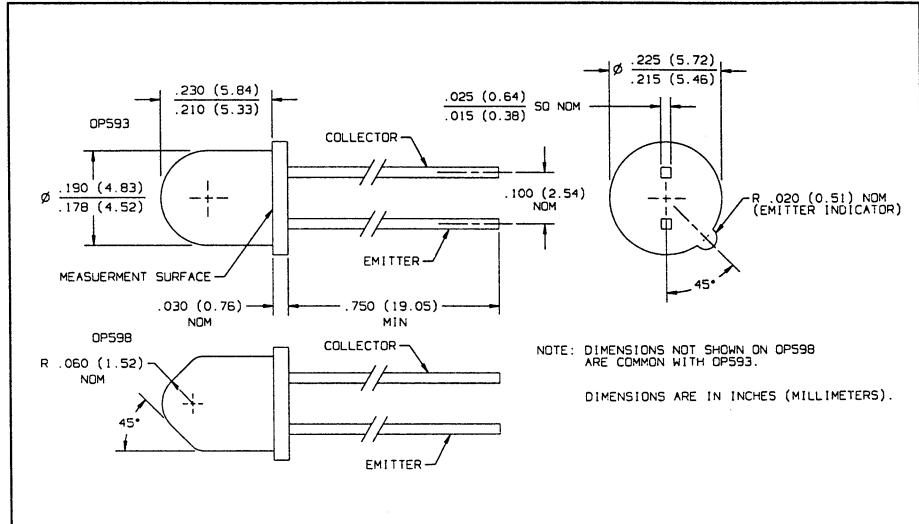
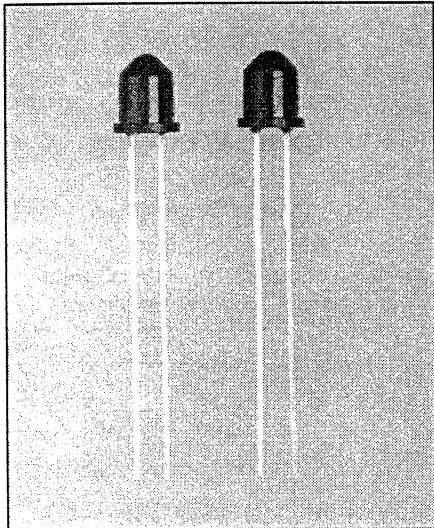
**ASIA/PACIFIC:** Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,  
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298



LM339/D



# NPN Plastic Silicon Phototransistors Types OP593, OP598 Series



## Features

- Wide receiving angle
- Variety of sensitivity ranges
- TO-18 equivalent package style

## Description

The OP593/598 series consist of NPN silicon phototransistors molded in dark blue epoxy packages. The wide receiving angle provides relatively even reception over a large area. These devices are 100% production tested using infrared light for close correlation with Optek's GaAs and GaAlAs emitters.

## Absolute Maximum Ratings ( $T_A = 25^\circ C$ unless otherwise noted)

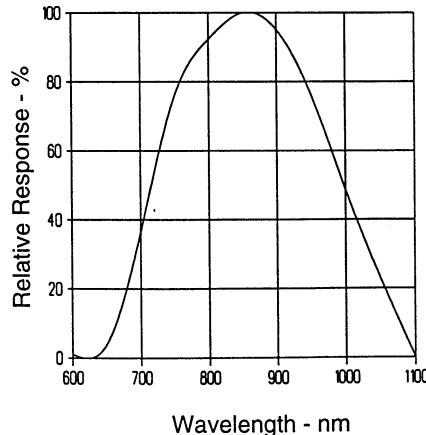
Collector-Emitter Voltage .....	30 V
Emitter-Collector Voltage .....	5.0 V
Continuous Collector Current .....	50 mA
Storage and Operating Temperature Range .....	-40° C to +100° C
Lead Soldering Temperature [1/16 inch (1.6 mm) from case for 5 sec. with soldering iron] .....	260° C <sup>(1)</sup>
Power Dissipation .....	250 mW <sup>(2)</sup>

### Notes:

- (1) RMA flux is recommended. Duration can be extended to 10 sec. max. when flow soldering. Max. 20 grams force may be applied to leads when soldering.
- (2) Derate linearly 3.33 mW/ $^\circ C$  above 25° C.
- (3)  $V_{CE} = 5$  V. Light source is an unfiltered GaAlAs emitting diode operating at peak emission wavelength of 890 nm and  $E_{e(APT)}$  of 1.7 mW/cm<sup>2</sup> average within a .250" dia. aperture.
- (4) This dimension is held to within  $\pm 0.005$ " on the flange edge and may vary up to  $\pm 0.020$ " in the area of the leads.

## Typical Performance Curves

### Typical Spectral Response



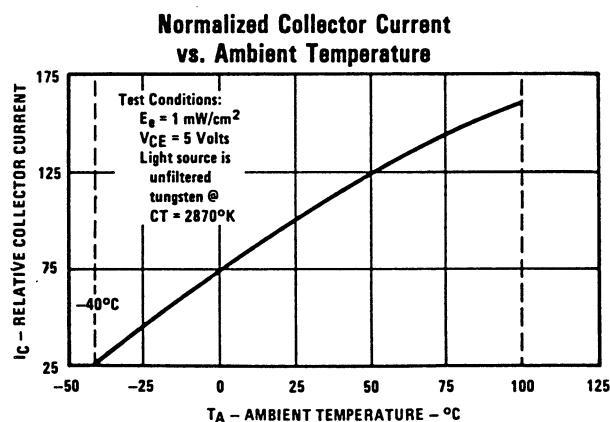
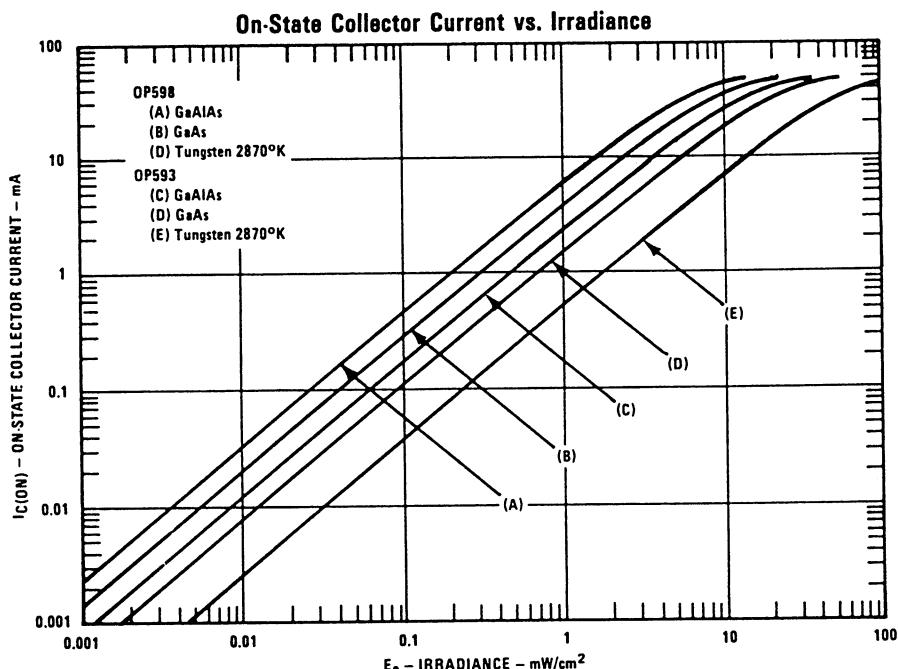
# Types OP593, OP598 Series

Electrical Characteristics ( $T_A = 25^\circ C$  unless otherwise noted)

SYMBOL	PARAMETER		MIN	TYP	MAX	UNITS	TEST CONDITIONS
I <sub>c(ON)</sub>	On-State Collector Current	OP593C	1.0			mA	See Note (3)
		OP593B	2.0			mA	
		OP593A	3.0			mA	
		OP598C	2.5			mA	See Note (3)
		OP598B	5.0			mA	
		OP598A	7.5			mA	
I <sub>CEO</sub>	Collector Dark Current				100	nA	$V_{CE} = 10 V, E_e = 0$
V <sub>(BR)CEO</sub>	Collector-Emitter Breakdown Voltage	30				V	$I_C = 100 \mu A$
V <sub>(BR)ECO</sub>	Emitter-Collector Breakdown Voltage	5				V	$I_E = 100 \mu A$
V <sub>CE(SAT)</sub>	Collector-Emitter Saturation Voltage				0.40	V	$I_C = 0.4 mA, E_e = 1.7 mW/cm^2(3)$

PHOTODIODES

## Typical Performance Curves

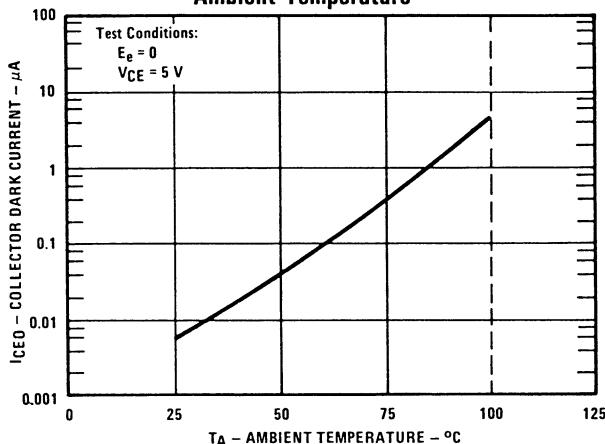


# Types OP593, OP598 Series

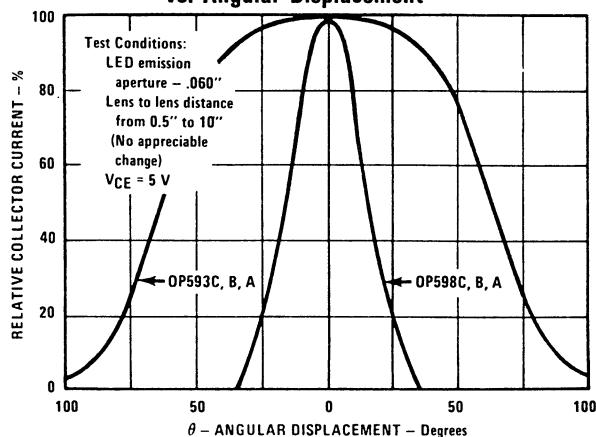


## Typical Performance Curves

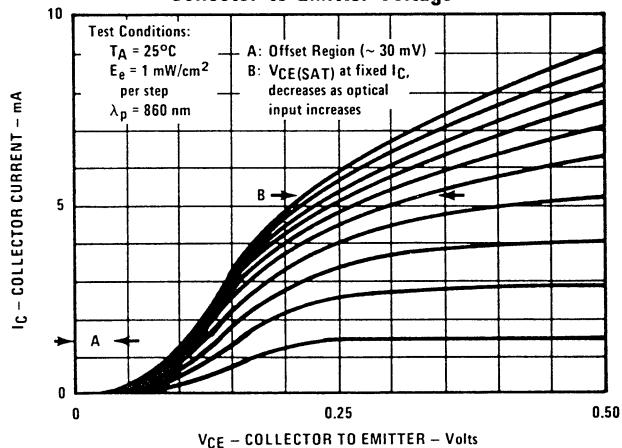
Collector Dark Current vs.  
Ambient Temperature



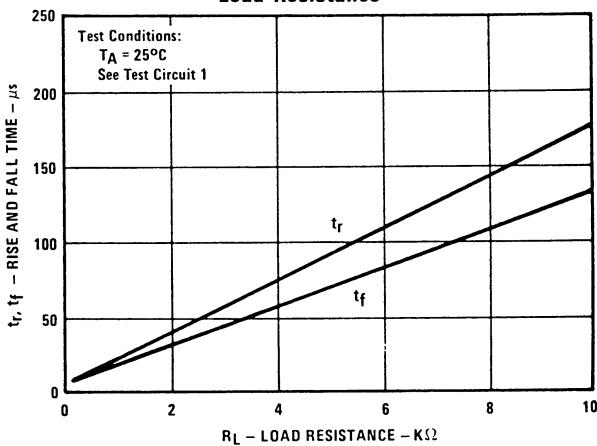
Relative Collector Current  
vs. Angular Displacement



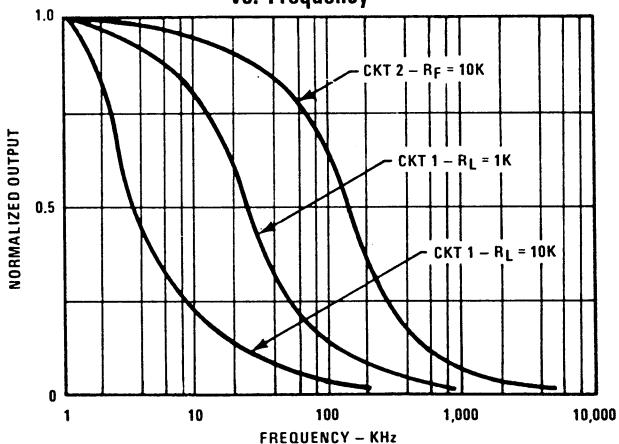
Collector Current vs.  
Collector to Emitter Voltage



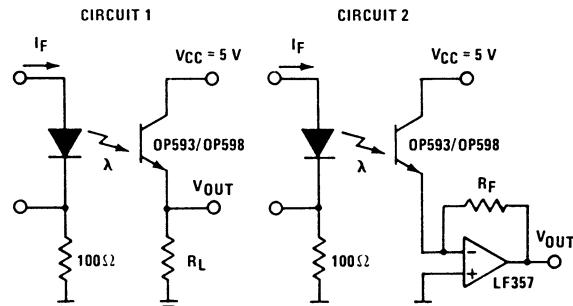
Rise and Fall Time vs.  
Load Resistance



Normalized Output  
vs. Frequency



Switching Time  
Test Circuit

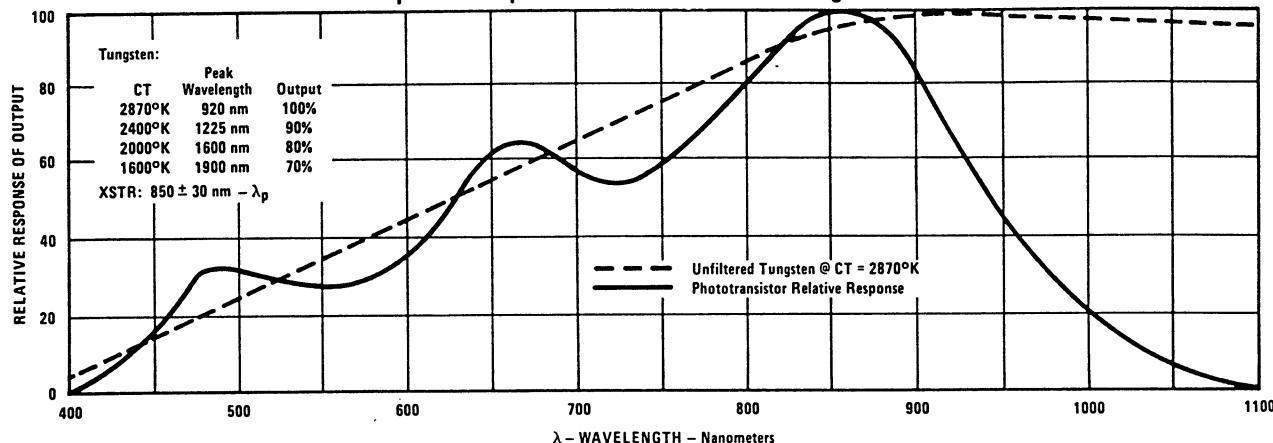


Test Conditions:  
Light source is pulsed LED with  $t_r$  and  $t_f \leq 500 \text{ ns}$ .  
 $I_F$  is adjusted for  $V_{OUT} = 1 \text{ Volt}$ .

# Types OP593, OP598 Series

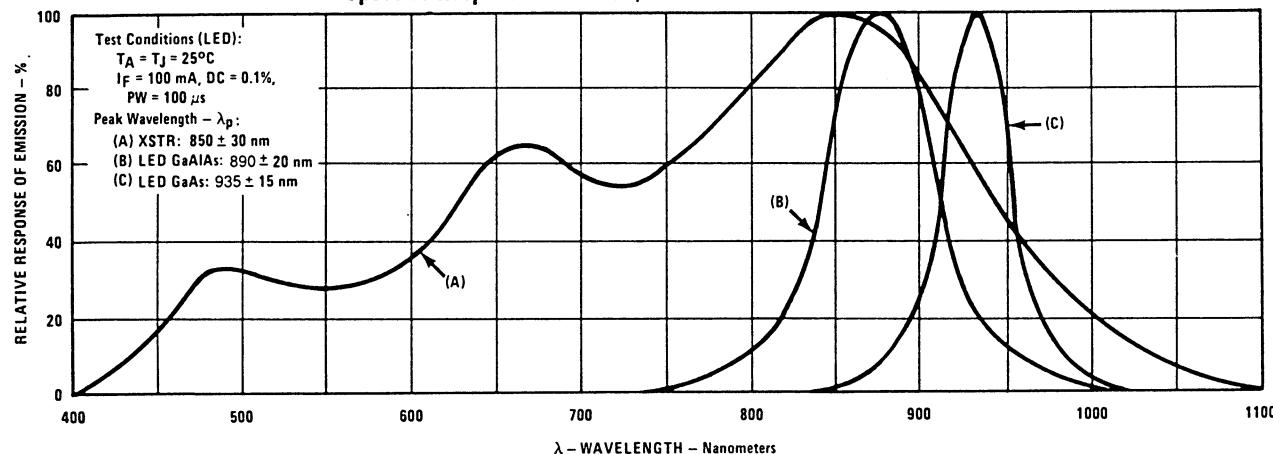
## Typical Performance Curves

Spectral Response of OP593/OP598 vs. Tungsten

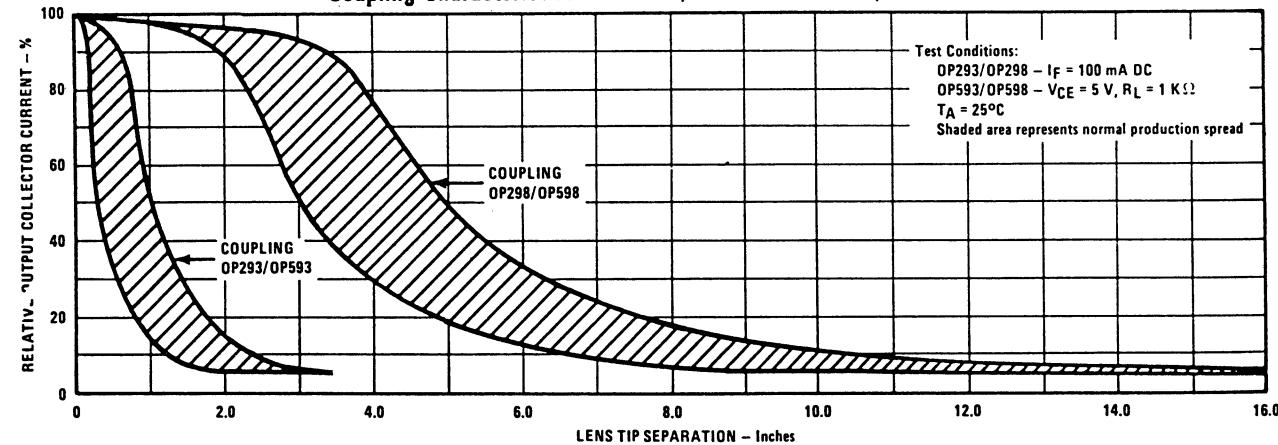


PHOTODIODES

Spectral Response of OP593/OP598 vs. GaAlAs and GaAs



Coupling Characteristics of OP293/OP593 and OP298/OP598



# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[TT Electronics:](#)

[OP598C](#) [OP593A](#)

# PS2501A-1, PS2501AL-1

HIGH ISOLATION VOLTAGE SINGLE TRANSISTOR TYPE

R08DS0209EJ0100

Rev.1.00

Dec 25, 2020

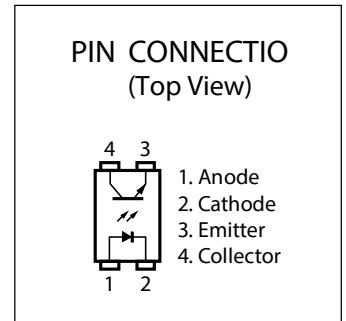
## DESCRIPTION

The PS2501A-1 and PS2501AL-1 are optically coupled isolators containing a GaAs light emitting diode and an NPN silicon phototransistor to realize an excellent cost performance.

The PS2501A-1 is in a plastic DIP (Dual In-line Package) and the PS2501AL-1 is lead bending type (Gull-wing) for surface mount.

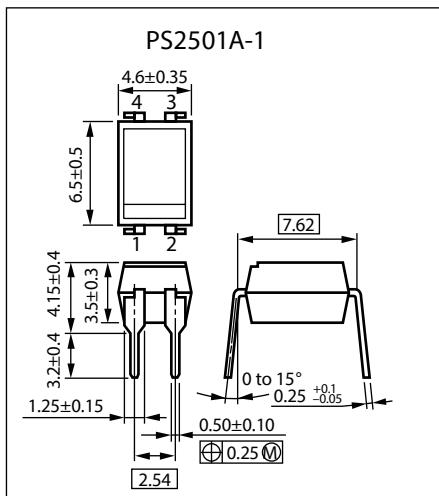
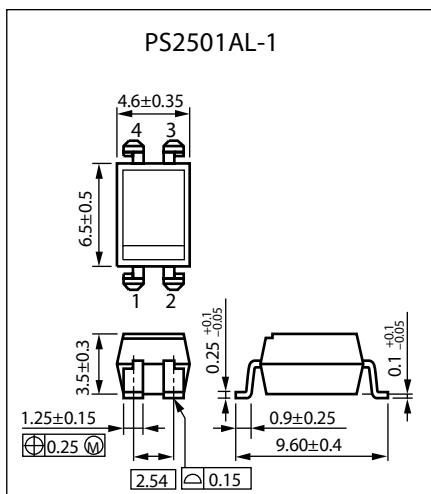
## FEATURES

- High isolation voltage ( $BV = 5\,000\text{ Vr.m.s.}$ )
- Ordering number of taping product: PS2501AL-1-F3 : 2 000 pcs/reel
- Pb-Free product
- Safety standards
  - UL approved: UL1577, Double protection



## APPLICATIONS

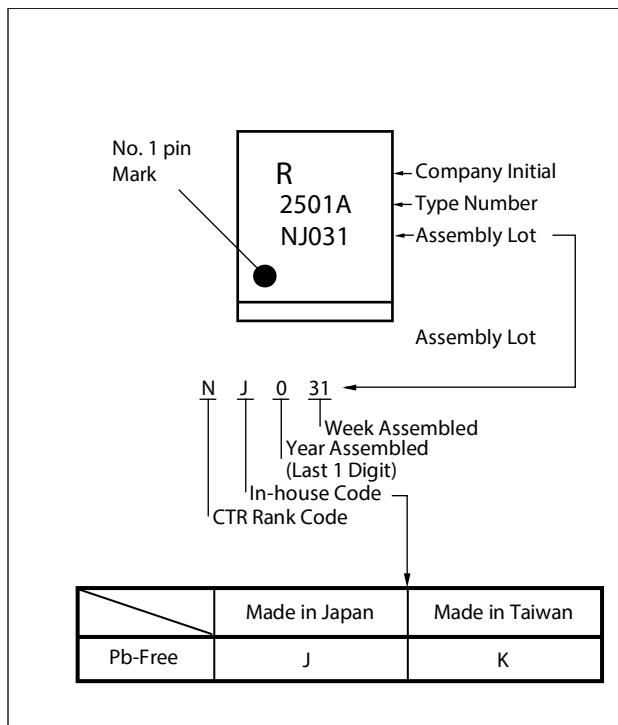
- Power supply
- Telephone/FAX.
- FA/OA equipment
- Programmable logic controller

**PACKAGE DIMENSIONS (UNIT: mm)****DIP Type****Lead Bending Type For Surface Mount**

Weight (4-pin DIP) : 0.26 g (typ.)

**PHOTOCOUPLED CONSTRUCTION**

Parameter	PS2501A-1, PS2501AL-1
Air Distance (MIN.)	7 mm
Creepage Distance (MIN.)	7 mm
Isolation Distance (MIN.)	0.3 mm

**MARKING EXAMPLE****ORDERING INFORMATION**

Part Number	Order Number <sup>*1</sup>	Solder Plating Specification	Packing Style	Safety Standard Approval	Application Part Number <sup>*2</sup>	
PS2501A-1	PS2501A-1-A	Pb-Free	Magazine case 100 pcs	Standard products (UL Approved)	PS2501A-1	
PS2501AL-1	PS2501AL-1-A					PS2501AL-1
PS2501AL-1-F3	PS2501AL-1-F3-A				Embossed Tape 2 000 pcs/reel	

Notes: \*1. When specifying CTR rank, please add “/CTR rank” after Order Number.

ex. L rank : PS2501A-1-A/L

Notes: \*2. For the application of the Safety Standard, following part number should be used.

**ABSOLUTE MAXIMUM RATINGS (T<sub>A</sub> = 25 °C, unless otherwise specified)**

Parameter		Symbol	Ratings	Unit
Diode	Reverse Voltage	V <sub>R</sub>	6	V
	Forward Current (DC)	I <sub>F</sub>	30	mA
	Power Dissipation Derating	ΔP <sub>D</sub> /°C	1.5	mW/°C
	Power Dissipation	P <sub>D</sub>	150	mW
	Peak Forward Current <sup>*1</sup>	I <sub>FP</sub>	0.5	A
Transistor	Collector to Emitter Voltage	V <sub>C EO</sub>	70	V
	Emitter to Collector Voltage	V <sub>E CO</sub>	5	V
	Collector Current	I <sub>C</sub>	30	mA
	Power Dissipation Delay	ΔP <sub>C</sub> /°C	1.5	mW/°C
	Power Dissipation	P <sub>C</sub>	150	mW
Isolation Voltage <sup>*2</sup>		BV	5 000	Vr.m.s.
Operating Ambient Temperature		T <sub>A</sub>	-55 to +100	°C
Storage Temperature		T <sub>stg</sub>	-55 to +150	°C

Note: \*1. PW = 100 μs, Duty Cycle = 1 %

\*2. AC voltage for 1 minute at T<sub>A</sub> = 25 °C, RH = 60 % between input and output.

Pins 1-2 shorted together, 3-4 shorted together.

ELECTRICAL CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ )

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Diode	Forward Voltage	$V_F$	$I_F = 10 \text{ mA}$		1.2	1.4	V
	Reverse Current	$I_R$	$V_R = 5 \text{ V}$			5	$\mu\text{A}$
	Terminal Capacitance	$C_t$	$V = 0 \text{ V}, f = 1.0 \text{ MHz}$		10		pF
Transistor	Collector to Emitter Dark Current	$I_{CEO}$	$V_{CE} = 70 \text{ V}, I_F = 0 \text{ mA}$			100	nA
Coupled	Current Transfer Ratio ( $I_C/I_F$ ) *1	CTR	$I_F = 5 \text{ mA}, V_{CE} = 5 \text{ V}$	50		400	%
	Collector Saturation Voltage	$V_{CE(\text{sat})}$	$I_F = 10 \text{ mA}, I_C = 2 \text{ mA}$		0.13	0.3	V
	Isolation Resistance	$R_{I-O}$	$V_{I-O} = 1.0 \text{ kV}_{\text{DC}}$	$10^{11}$			$\Omega$
	Isolation Capacitance	$C_{I-O}$	$V = 0 \text{ V}, f = 1.0 \text{ MHz}$		0.4		pF
	Rise Time*2	$t_r$	$V_{CC} = 10 \text{ V}, I_C = 2 \text{ mA}, R_L = 100 \Omega$		3		$\mu\text{s}$
	Fall Time*2	$t_f$			5		

Note : \*1. CTR rank

N : 50 to 400 (%)

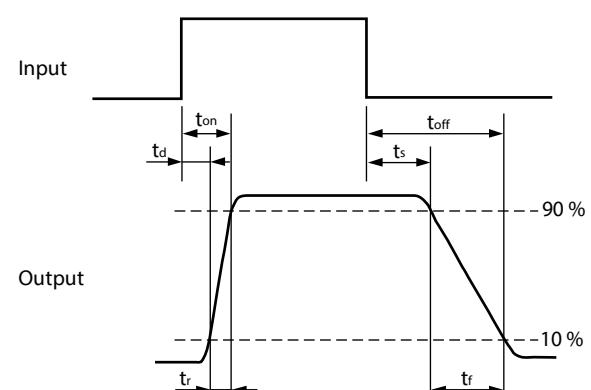
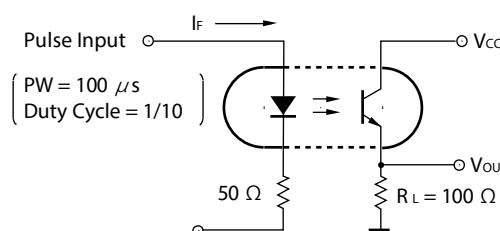
H : 80 to 160 (%)

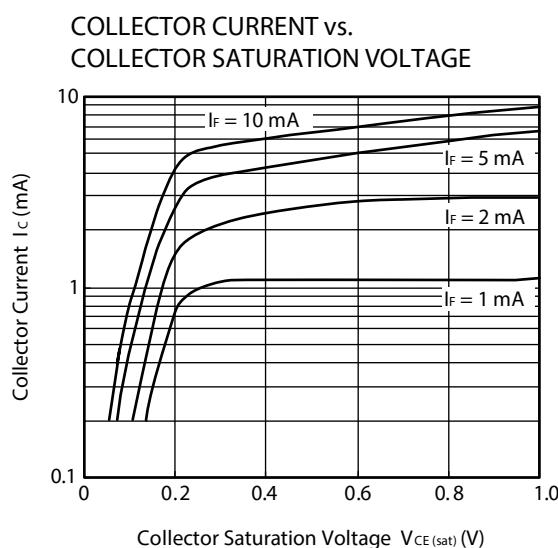
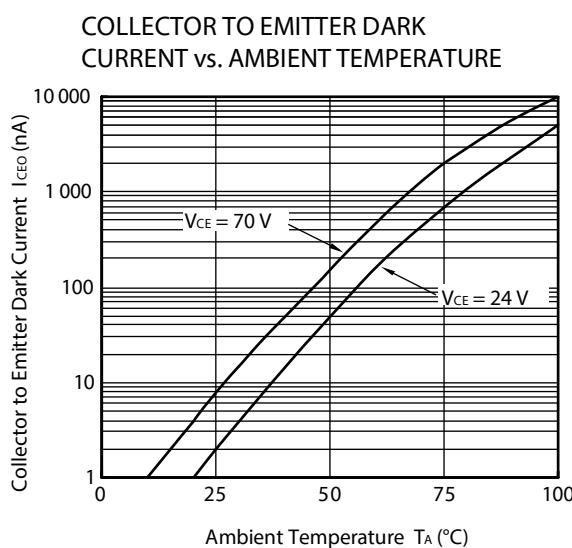
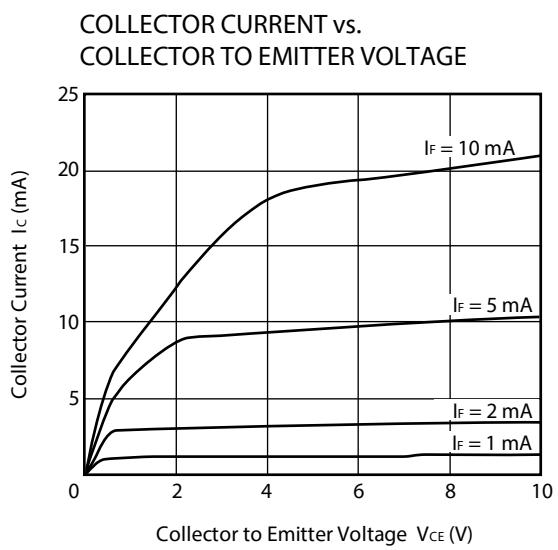
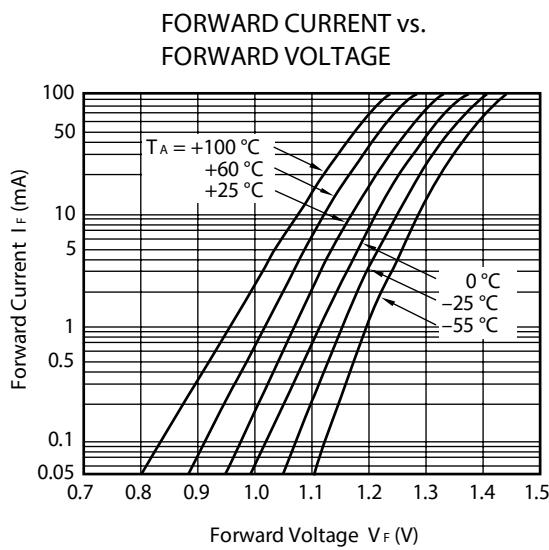
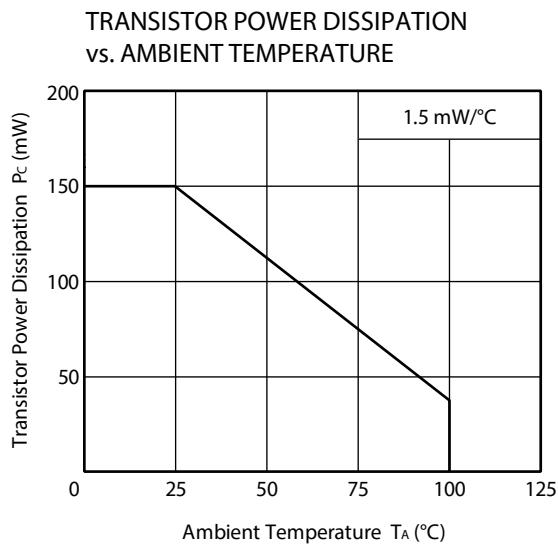
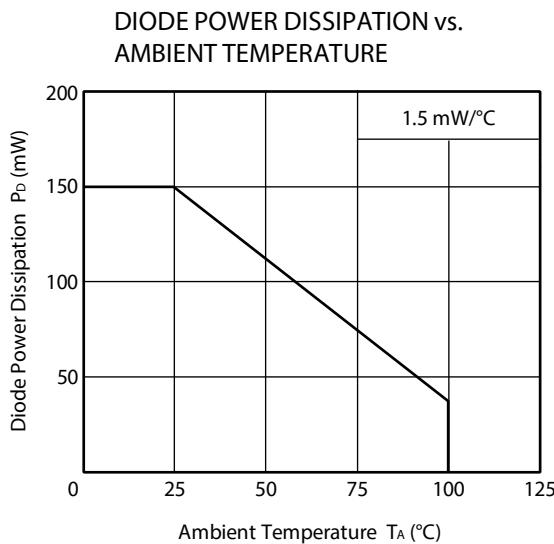
W : 130 to 260 (%)

Q : 100 to 200 (%)

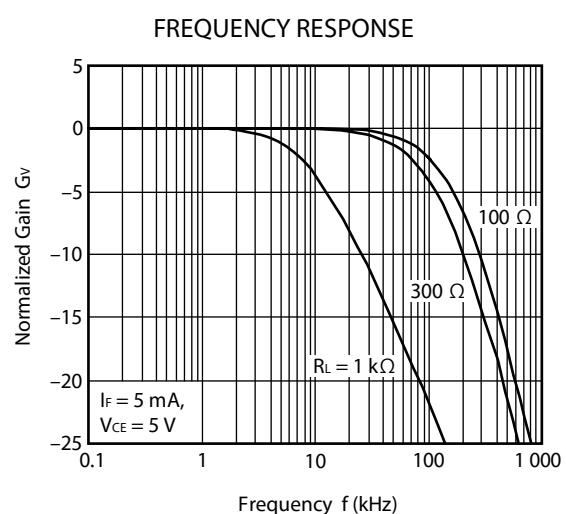
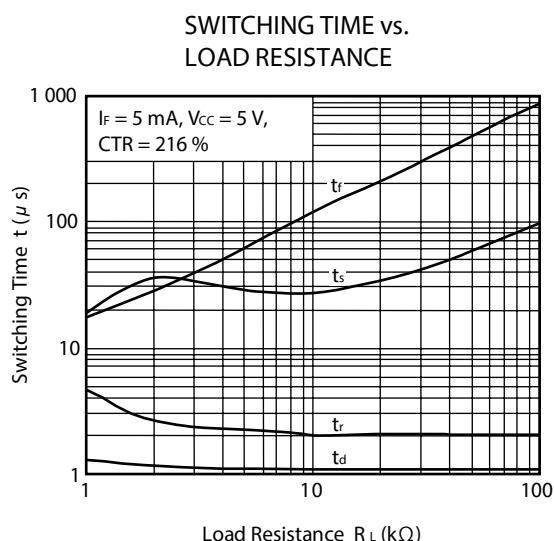
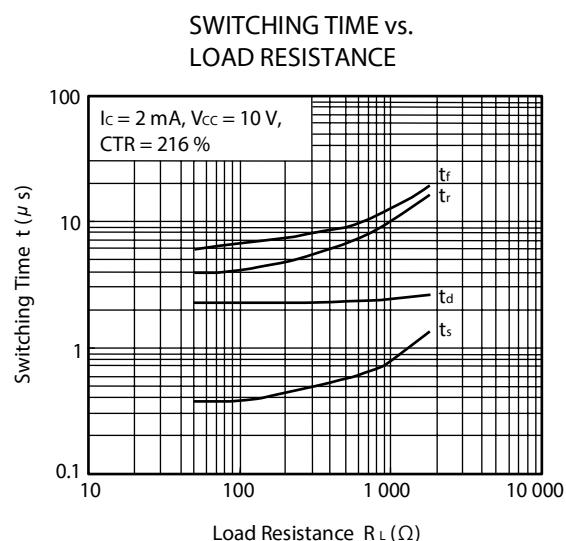
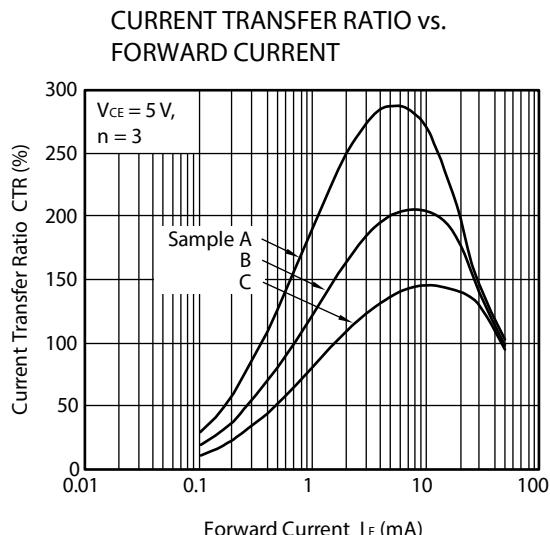
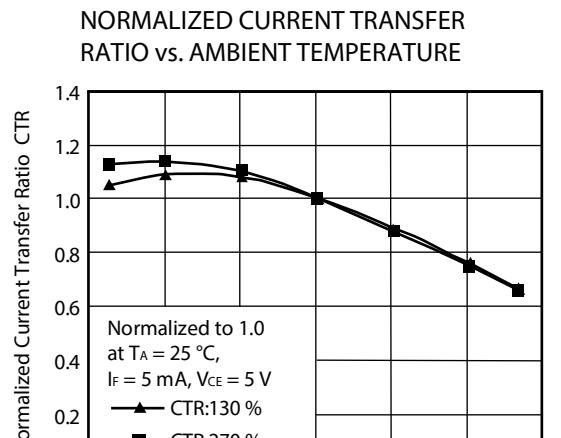
L : 200 to 400 (%)

\*2. Test Circuit for Switching Time



**TYPICAL CHARACTERISTICS (T<sub>A</sub> = 25 °C, unless otherwise specified)**

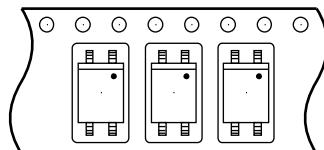
**Remark** The graphs indicate nominal characteristics.



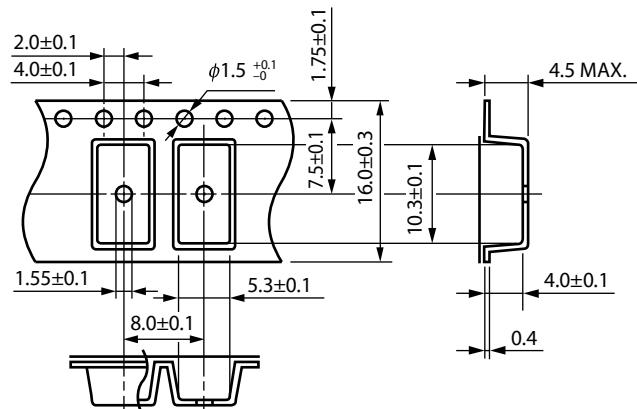
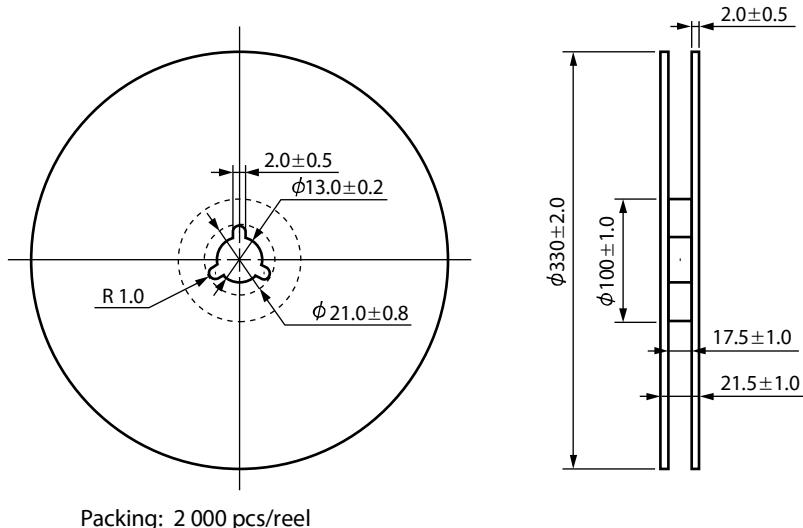
**Remark** The graphs indicate nominal characteristics.

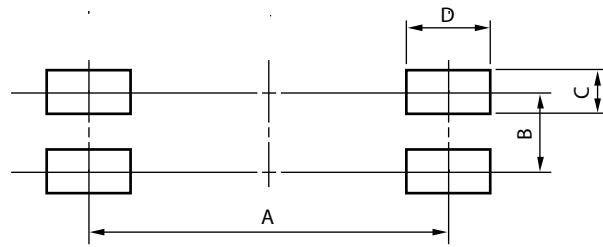
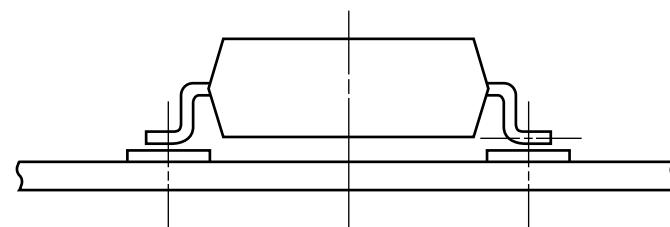
**TAPING SPECIFICATIONS (UNIT: mm)****Taping Direction**

PS2501AL-1-F3



Direction of feed

**Outline and Dimensions (Tape)****Outline and Dimensions (Reel)**

**RECOMMENDED MOUNT PAD DIMENSIONS (UNIT: mm)**

Part Number	Lead Bending	A	B	C	D
PS2501AL	Lead Bending Type For Surface Mount	8.2	2.54	1.7	2.2

**Remark** All dimensions in this figure must be evaluated before use.

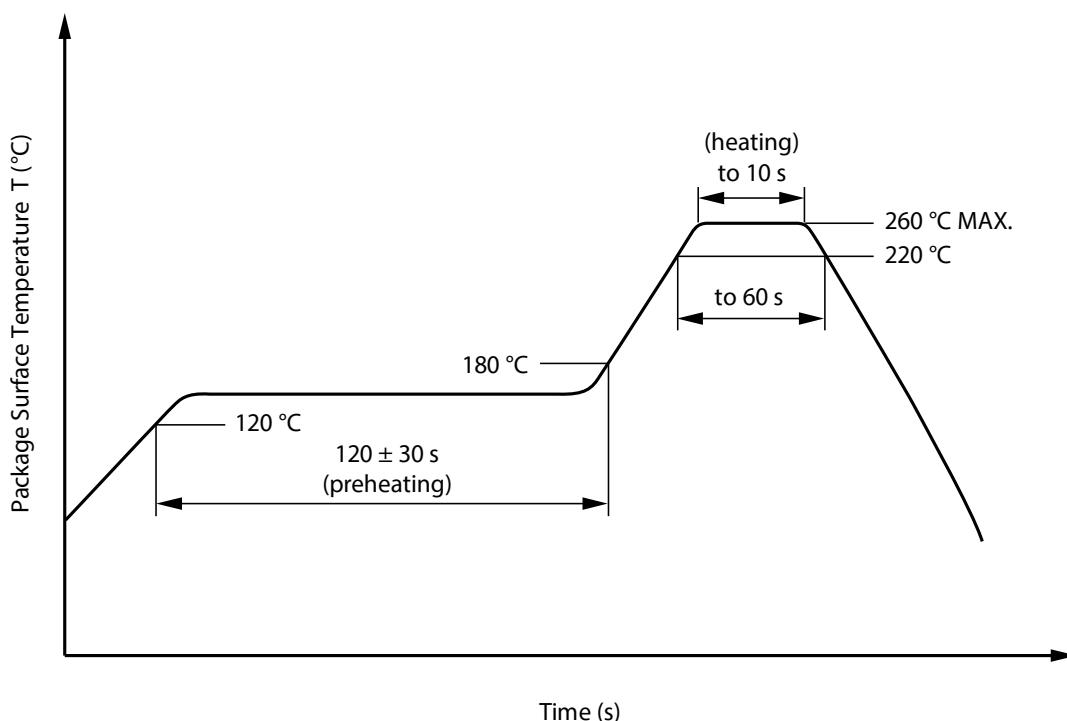
## NOTES ON HANDLING

### 1. Recommended soldering conditions

#### (1) Infrared reflow soldering

• Peak reflow temperature	260 °C or below (package surface temperature)
• Time of peak reflow temperature	10 seconds or less
• Time of temperature higher than 220°C	60 seconds or less
• Time to preheat temperature from 120 to 180°C	120 ± 30 s
• Number of reflows	Three
• Flux	Rosin flux containing small amount of chlorine (The flux with a maximum chlorine content of 0.2 Wt% is recommended.)

Recommended Temperature Profile of Infrared Reflow



#### (2) Wave soldering

• Temperature	260 °C or below (molten solder temperature)
• Time	10 seconds or less
• Preheating conditions	120 °C or below (package surface temperature)
• Number of times	One (Allowed to be dipped in solder including plastic mold portion.)
• Flux	Rosin flux containing small amount of chlorine (The flux with a maximum chlorine content of 0.2 Wt% is recommended.)

#### (3) Soldering by Soldering Iron

• Peak Temperature (lead part temperature)	350 °C or below
• Time (each pins)	3 seconds or less
• Flux	Rosin flux containing small amount of chlorine (The flux with a maximum chlorine content of 0.2 Wt% is recommended.)

- (a) Soldering of leads should be made at the point 1.5 to 2.0 mm from the root of the lead
- (b) Please be sure that the temperature of the package would not be heated over 100 °C

#### (4) Cautions

- Flux Cleaning
  - Avoid cleaning with Freon based or halogen-based (chlorinated etc.) solvents.
  - Do not use fixing agents or coatings containing halogen-based substances.

**2. Cautions regarding noise**

Be aware that when voltage is applied suddenly between the photocoupler's input and output or between collector-emitters at startup, the output transistor may enter the on state, even if the voltage is within the absolute maximum ratings.

**3. Measurement conditions of current transfer ratios (CTR), which differ according to photocoupler**

Check the setting values before use, since the forward current conditions at CTR measurement differ according to product.

When using products other than at the specified forward current, the characteristics curves may differ from the standard curves due to CTR value variations or the like. This tendency may sometimes be obvious, especially below  $I_F = 1$  mA.

Therefore, check the characteristics under the actual operating conditions and thoroughly take variations or the like into consideration before use.

## **USAGE CAUTIONS**

1. Protect against static electricity when handling.
2. Avoid storage at a high temperature and high humidity.
3. Avoid cleaning with Freon based or halogen-based (chlorinated etc.) solvents.
4. Do not use fixing agents or coatings containing halogen-based substances.

<b>Caution</b>	<p>GaAs Products</p> <p>This product uses gallium arsenide (GaAs). GaAs vapor and powder are hazardous to human health if inhaled or ingested, so please observe the following points.</p> <ul style="list-style-type: none"><li>• Follow related laws and ordinances when disposing of the product. If there are no applicable laws and/or ordinances, dispose of the product as recommended below.</li><li>1. Commission a disposal company able to (with a license to) collect, transport and dispose of materials that contain arsenic and other such industrial waste materials.</li><li>2. Exclude the product from general industrial waste and household garbage, and ensure that the product is controlled (as industrial waste subject to special control) up until final disposal.</li><li>• Do not burn, destroy, cut, crush, or chemically dissolve the product.</li><li>• Do not lick the product or in any way allow it to enter the mouth.</li></ul>
----------------	--

All trademarks and registered trademarks are the property of their respective owners.

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
  2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
  3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
  4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
  5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc. Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
  6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
  7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
  8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
  9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
  10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
  11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
  12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)



### SALES OFFICES

### Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

#### Renesas Electronics Corporation

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

#### Renesas Electronics America Inc.

1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.

Tel: +1-408-432-8888, Fax: +1-408-434-5351

#### Renesas Electronics Canada Limited

9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3

Tel: +1-905-237-2004

#### Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany

Tel: +49-211-6503-0, Fax: +49-211-6503-1327

#### Renesas Electronics (China) Co., Ltd.

Room 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, China

Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

#### Renesas Electronics (Shanghai) Co., Ltd.

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai 200333, China

Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

#### Renesas Electronics Hong Kong Limited

Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong

Tel: +852-2265-6688, Fax: +852 2886-9022

#### Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan

Tel: +886 2-8175-9600, Fax: +886 2-8175-9670

#### Renesas Electronics Singapore Pte. Ltd.

80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949

Tel: +65-6213-0200, Fax: +65-6213-0300

#### Renesas Electronics Malaysia Sdn.Bhd.

Unit No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia

Tel: +60-3-5022-1288, Fax: +60-3-5022-1290

#### Renesas Electronics India Pvt. Ltd.

No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India

Tel: +91-80-67208700

#### Renesas Electronics Korea Co., Ltd.

17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea

Tel: +82-2-558-3737, Fax: +82-2-558-5338

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Renesas Electronics:](#)

[PS2501AL-1-F3-H-A](#) [PS2501AL-1-F3-A](#) [PS2501A-1-A](#) [PS2501AL-1-A](#)

# DATA SHEET

## **PCF8574**

Remote 8-bit I/O expander for  
I<sup>2</sup>C-bus

Product specification

1997 Apr 02

Supersedes data of September 1994

File under Integrated Circuits, IC12

**Remote 8-bit I/O expander for I<sup>2</sup>C-bus****PCF8574****CONTENTS**

- 1 FEATURES
- 2 GENERAL DESCRIPTION
- 3 ORDERING INFORMATION
- 4 BLOCK DIAGRAM
- 5 PINNING
- 6 CHARACTERISTICS OF THE I<sup>2</sup>C-BUS
  - 6.1 Bit transfer
  - 6.2 Start and stop conditions
  - 6.3 System configuration
  - 6.4 Acknowledge
- 7 FUNCTIONAL DESCRIPTION
  - 7.1 Addressing
  - 7.2 Interrupt
  - 7.3 Quasi-bidirectional I/Os
- 8 LIMITING VALUES
- 9 HANDLING
- 10 DC CHARACTERISTICS
- 11 I<sup>2</sup>C-BUS TIMING CHARACTERISTICS
- 12 PACKAGE OUTLINES
- 13 SOLDERING
  - 13.1 Introduction
  - 13.2 DIP
    - 13.2.1 Soldering by dipping or by wave
    - 13.2.2 Repairing soldered joints
  - 13.3 SO and SSOP
    - 13.3.1 Reflow soldering
    - 13.3.2 Wave soldering
    - 13.3.3 Repairing soldered joints
- 14 DEFINITIONS
- 15 LIFE SUPPORT APPLICATIONS
- 16 PURCHASE OF PHILIPS I<sup>2</sup>C COMPONENTS



## Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

### 1 FEATURES

- Operating supply voltage 2.5 to 6 V
- Low standby current consumption of 10 µA maximum
- I<sup>2</sup>C to parallel port expander
- Open-drain interrupt output
- 8-bit remote I/O port for the I<sup>2</sup>C-bus
- Compatible with most microcontrollers
- Latched outputs with high current drive capability for directly driving LEDs
- Address by 3 hardware address pins for use of up to 8 devices (up to 16 with PCF8574A)
- DIP16, or space-saving SO16 or SSOP20 packages.

### 2 GENERAL DESCRIPTION

The PCF8574 is a silicon CMOS circuit. It provides general purpose remote I/O expansion for most microcontroller families via the two-line bidirectional bus (I<sup>2</sup>C).

The device consists of an 8-bit quasi-bidirectional port and an I<sup>2</sup>C-bus interface. The PCF8574 has a low current consumption and includes latched outputs with high current drive capability for directly driving LEDs. It also possesses an interrupt line (INT) which can be connected to the interrupt logic of the microcontroller. By sending an interrupt signal on this line, the remote I/O can inform the microcontroller if there is incoming data on its ports without having to communicate via the I<sup>2</sup>C-bus. This means that the PCF8574 can remain a simple slave device.

The PCF8574 and PCF8574A versions differ only in their slave address as shown in Fig.9.

### 3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCF8574P; PCF8574AP	DIP16	plastic dual in-line package; 16 leads (300 mil)	SOT38-1
PCF8574T; PCF8574AT	SO16	plastic small outline package; 16 leads; body width 7.5 mm	SOT162-1
PCF8574TS	SSOP20	plastic shrink small outline package; 20 leads; body width 4.4 mm	SOT266-1

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

## 4 BLOCK DIAGRAM

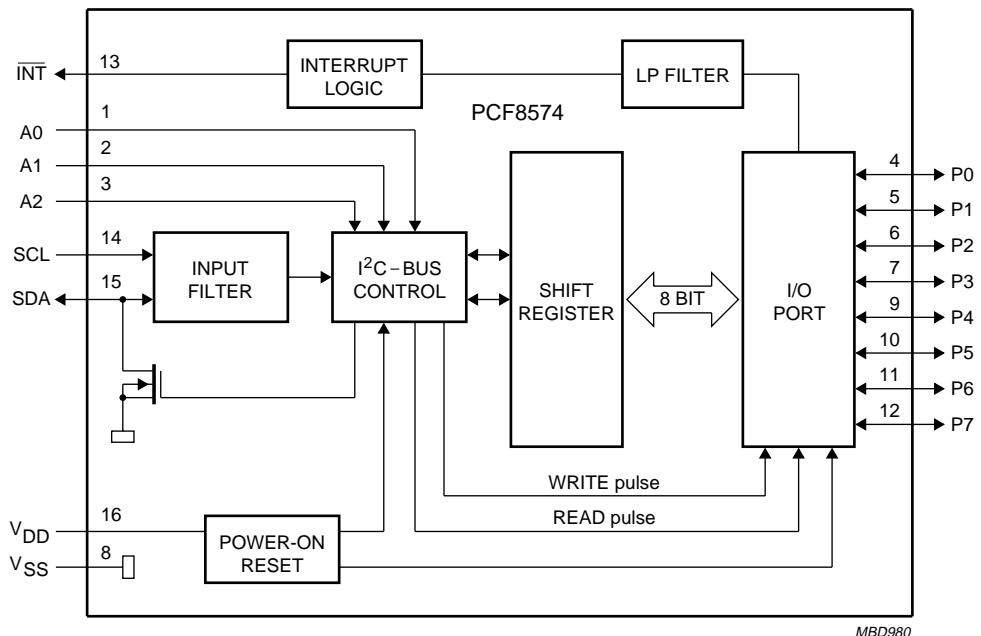


Fig.1 Block diagram (SOT38-1 and SOT162-1).

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

## 5 PINNING

SYMBOL	PIN		DESCRIPTION
	DIP16; SO16	SSOP20	
A0	1	6	address input 0
A1	2	7	address input 1
A2	3	9	address input 2
P0	4	10	quasi-bidirectional I/O 0
P1	5	11	quasi-bidirectional I/O 1
P2	6	12	quasi-bidirectional I/O 2
P3	7	14	quasi-bidirectional I/O 3
V <sub>SS</sub>	8	15	supply ground
P4	9	16	quasi-bidirectional I/O 4
P5	10	17	quasi-bidirectional I/O 5
P6	11	19	quasi-bidirectional I/O 6
P7	12	20	quasi-bidirectional I/O 7
INT	13	1	interrupt output (active LOW)
SCL	14	2	serial clock line
SDA	15	4	serial data line
V <sub>DD</sub>	16	5	supply voltage
n.c.	—	3	not connected
n.c.	—	8	not connected
n.c.	—	13	not connected
n.c.	—	18	not connected

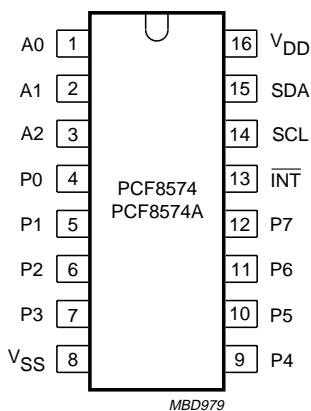


Fig.2 Pin configuration (DIP16; SO16).

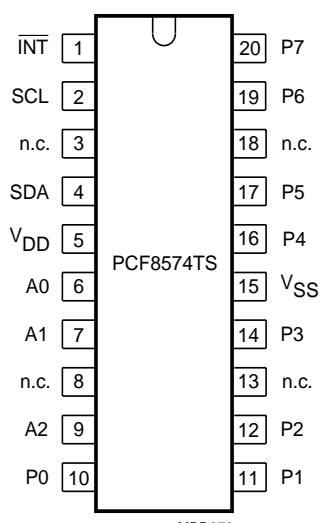


Fig.3 Pin configuration (SSOP20).

## Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

### 6 CHARACTERISTICS OF THE I<sup>2</sup>C-BUS

The I<sup>2</sup>C-bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

#### 6.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (see Fig.4).

#### 6.2 Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P) (see Fig.5).

#### 6.3 System configuration

A device generating a message is a 'transmitter', a device receiving is the 'receiver'. The device that controls the message is the 'master' and the devices which are controlled by the master are the 'slaves' (see Fig.6).

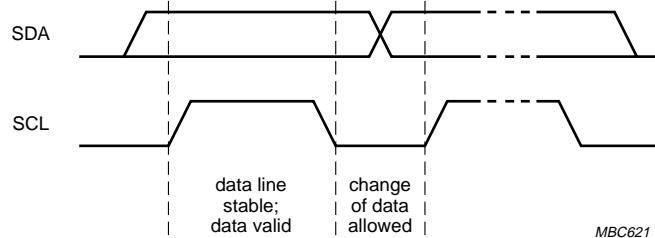


Fig.4 Bit transfer.

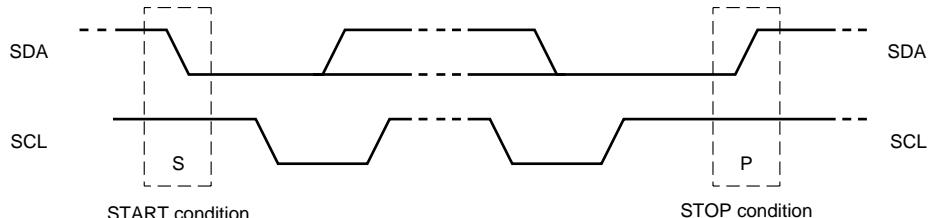


Fig.5 Definition of start and stop conditions.

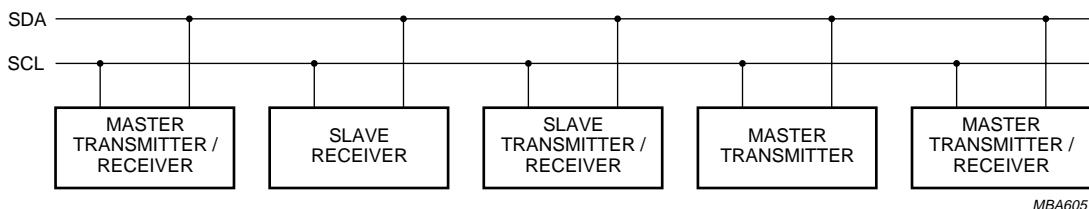


Fig.6 System configuration.

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

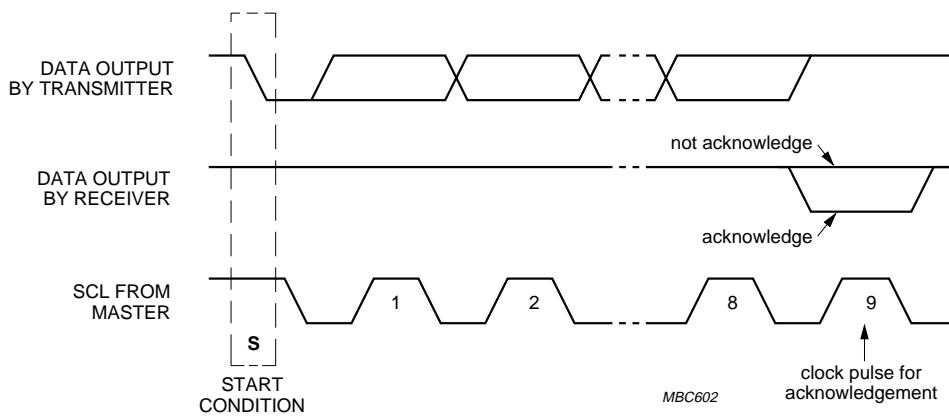
**6.4 Acknowledge**

The number of data bytes transferred between the start and the stop conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse.

A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave

transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set-up and hold times must be taken into account.

A master receiver must signal an end of data to the transmitter by **not** generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

Fig.7 Acknowledgment on the I<sup>2</sup>C-bus.

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

## 7 FUNCTIONAL DESCRIPTION

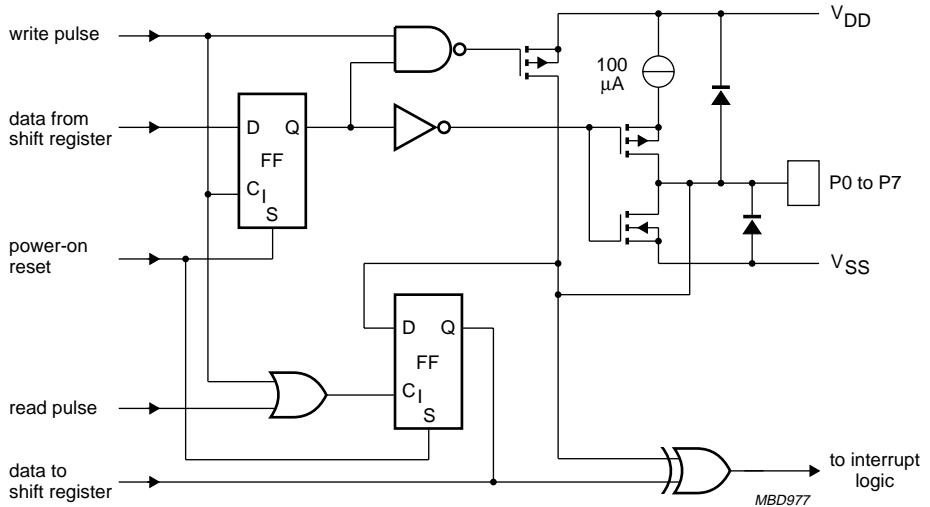
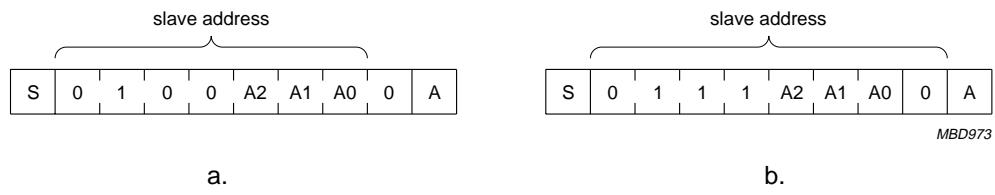


Fig.8 Simplified schematic diagram of each I/O.

## 7.1 Addressing

For addressing see Figs 9, 10 and 11.



- (a) PCF8574.  
 (b) PCF8574A.

Fig.9 PCF8574 and PCF8574A slave addresses.

Each of the PCF8574's eight I/Os can be independently used as an input or output. Input data is transferred from the port to the microcontroller by the READ mode (see Fig.11). Output data is transmitted to the port by the WRITE mode (see Fig.10).

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

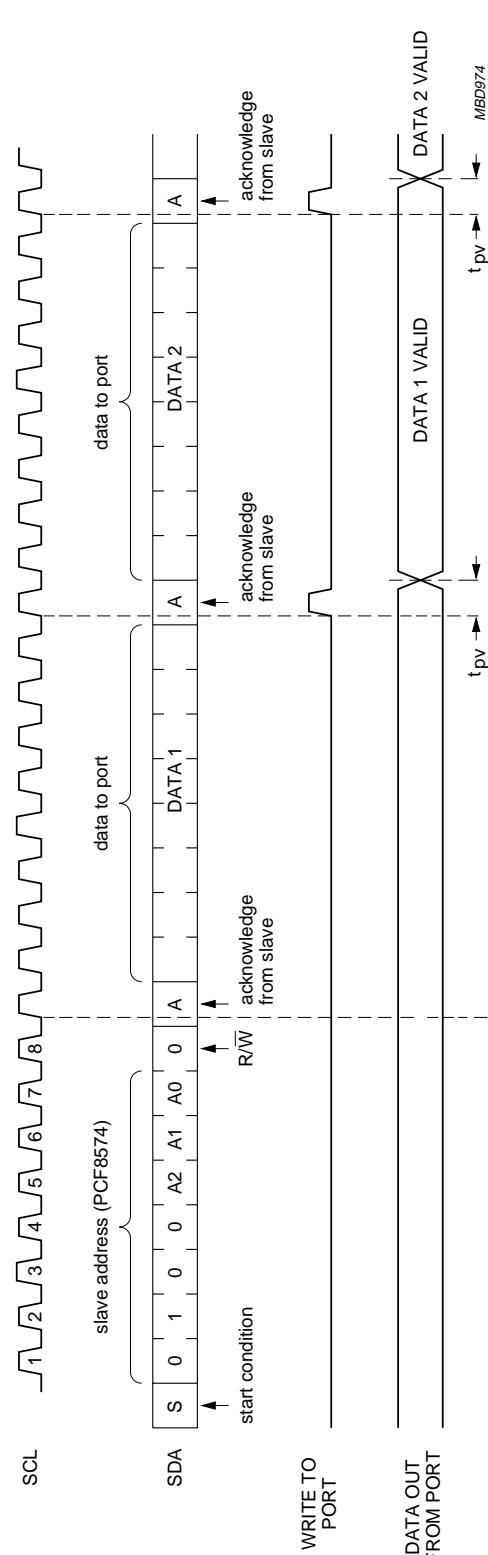
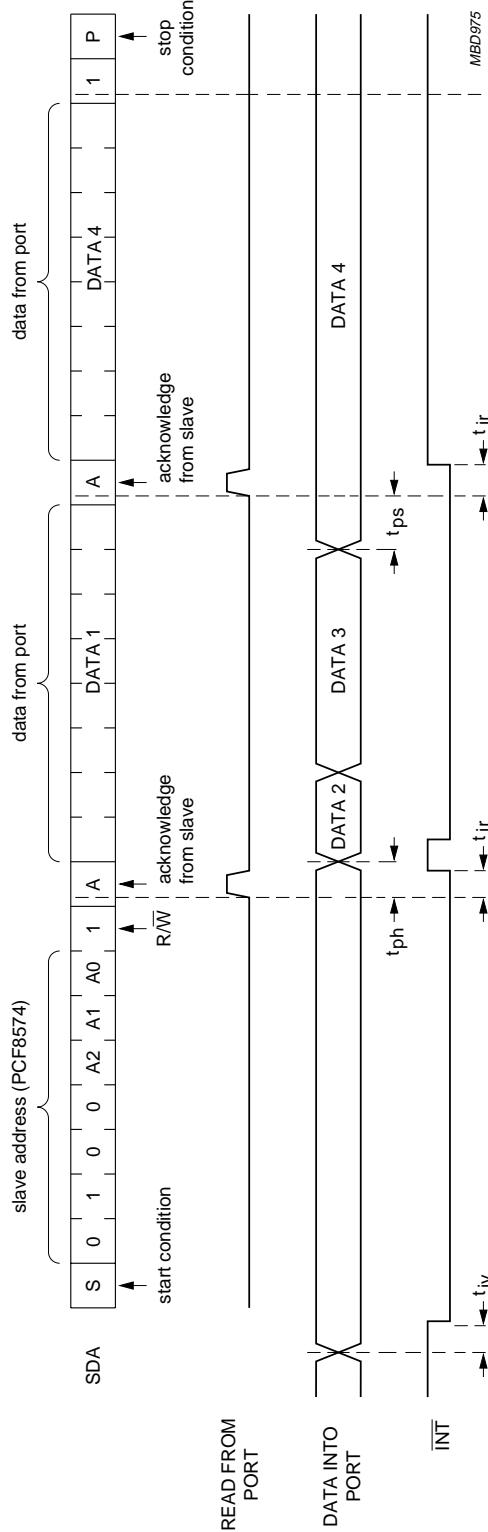


Fig.10 WRITE mode (output).

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574



A LOW-to-HIGH transition of SDA, while SCL is HIGH is defined as the stop condition (P). Transfer of data can be stopped at any moment by a stop condition. When this occurs, data present at the last acknowledge phase is valid (output mode). Input data is lost.

Fig.11 READ mode (input).

## Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

### 7.2 Interrupt (see Figs 12 and 13)

The PCF8574 provides an open drain output ( $\overline{\text{INT}}$ ) which can be fed to a corresponding input of the microcontroller. This gives these chips a type of master function which can initiate an action elsewhere in the system.

An interrupt is generated by any rising or falling edge of the port inputs in the input mode. After time  $t_{iv}$  the signal  $\overline{\text{INT}}$  is valid.

Resetting and reactivating the interrupt circuit is achieved when data on the port is changed to the original setting or data is read from or written to the port which has generated the interrupt.

Resetting occurs as follows:

- In the READ mode at the acknowledge bit after the rising edge of the SCL signal
- In the WRITE mode at the acknowledge bit after the HIGH-to-LOW transition of the SCL signal

- Interrupts which occur during the acknowledge clock pulse may be lost (or very short) due to the resetting of the interrupt during this pulse.

Each change of the I/Os after resetting will be detected and, after the next rising clock edge, will be transmitted as  $\overline{\text{INT}}$ . Reading from or writing to another device does not affect the interrupt circuit.

### 7.3 Quasi-bidirectional I/Os (see Fig.14)

A quasi-bidirectional I/O can be used as an input or output without the use of a control signal for data direction.

At power-on the I/Os are HIGH. In this mode only a current source to  $V_{DD}$  is active. An additional strong pull-up to  $V_{DD}$  allows fast rising edges into heavily loaded outputs. These devices turn on when an output is written HIGH, and are switched off by the negative edge of SCL. The I/Os should be HIGH before being used as inputs.

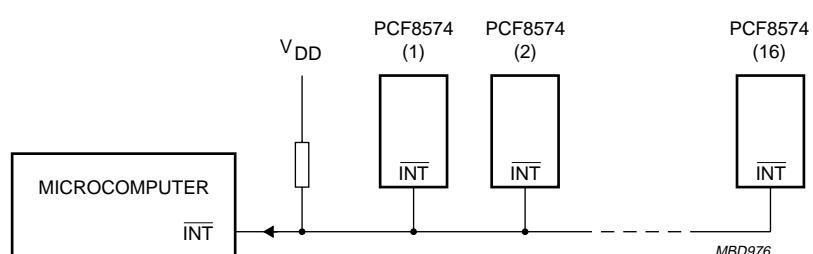


Fig.12 Application of multiple PCF8574s with interrupt.

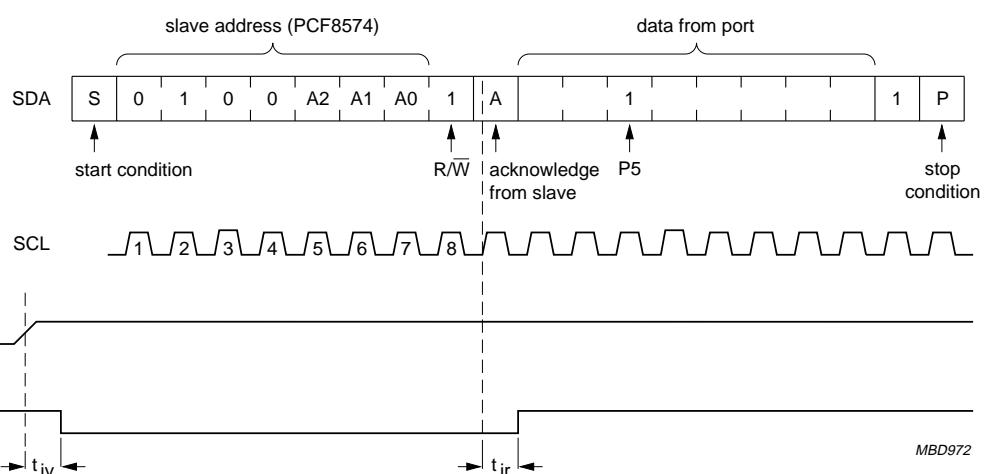
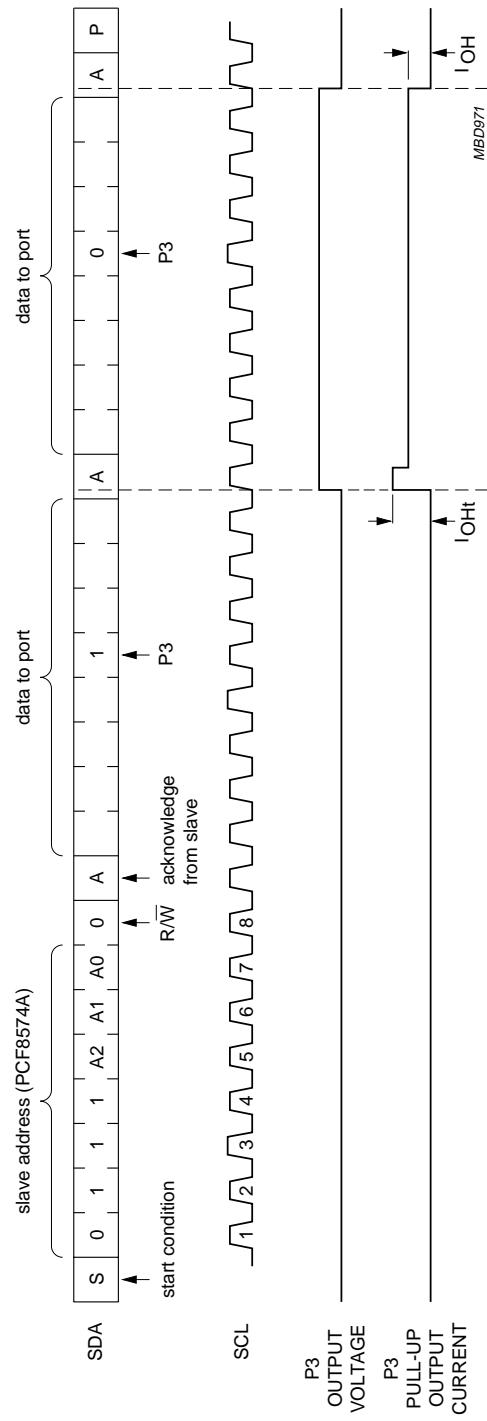


Fig.13 Interrupt generated by a change of input to I/O P5.

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

Fig.14 Transient pull-up current  $I_{OHt}$  while P3 changes from LOW-to-HIGH and back to LOW.

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

**8 LIMITING VALUES**

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
V <sub>DD</sub>	supply voltage	-0.5	+7.0	V
V <sub>I</sub>	input voltage	V <sub>SS</sub> - 0.5	V <sub>DD</sub> + 0.5	V
I <sub>I</sub>	DC input current	-	±20	mA
I <sub>O</sub>	DC output current	-	±25	mA
I <sub>DD</sub>	supply current	-	±100	mA
I <sub>SS</sub>	supply current	-	±100	mA
P <sub>tot</sub>	total power dissipation	-	400	mW
P <sub>O</sub>	power dissipation per output	-	100	mW
T <sub>stg</sub>	storage temperature	-65	+150	°C
T <sub>amb</sub>	operating ambient temperature	-40	+85	°C

**9 HANDLING**

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take precautions appropriate to handling MOS devices. Advice can be found in Data Handbook IC12 under "Handling MOS Devices".

**10 DC CHARACTERISTICS**V<sub>DD</sub> = 2.5 to 6 V; V<sub>SS</sub> = 0 V; T<sub>amb</sub> = -40 to +85 °C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
<b>Supply</b>						
V <sub>DD</sub>	supply voltage		2.5	-	6.0	V
I <sub>DD</sub>	supply current	operating mode; V <sub>DD</sub> = 6 V; no load; V <sub>I</sub> = V <sub>DD</sub> or V <sub>SS</sub> ; f <sub>SCL</sub> = 100 kHz	-	40	100	µA
I <sub>stb</sub>	standby current	standby mode; V <sub>DD</sub> = 6 V; no load; V <sub>I</sub> = V <sub>DD</sub> or V <sub>SS</sub>	-	2.5	10	µA
V <sub>POR</sub>	Power-on reset voltage	V <sub>DD</sub> = 6 V; no load; V <sub>I</sub> = V <sub>DD</sub> or V <sub>SS</sub> ; note 1	-	1.3	2.4	V
<b>Input SCL; input/output SDA</b>						
V <sub>IL</sub>	LOW level input voltage		-0.5	-	+0.3V <sub>DD</sub>	V
V <sub>IH</sub>	HIGH level input voltage		0.7V <sub>DD</sub>	-	V <sub>DD</sub> + 0.5	V
I <sub>OL</sub>	LOW level output current	V <sub>OL</sub> = 0.4 V	3	-	-	mA
I <sub>L</sub>	leakage current	V <sub>I</sub> = V <sub>DD</sub> or V <sub>SS</sub>	-1	-	+1	µA
C <sub>i</sub>	input capacitance	V <sub>I</sub> = V <sub>SS</sub>	-	-	7	pF

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
<b>I/Os</b>						
V <sub>IL</sub>	LOW level input voltage		-0.5	-	+0.3V <sub>DD</sub>	V
V <sub>IH</sub>	HIGH level input voltage		0.7V <sub>DD</sub>	-	V <sub>DD</sub> + 0.5	V
I <sub>IHL(max)</sub>	maximum allowed input current through protection diode	V <sub>I</sub> ≥ V <sub>DD</sub> or V <sub>I</sub> ≤ V <sub>SS</sub>	-	-	±400	µA
I <sub>OL</sub>	LOW level output current	V <sub>OL</sub> = 1 V; V <sub>DD</sub> = 5 V	10	25	-	mA
I <sub>OH</sub>	HIGH level output current	V <sub>OH</sub> = V <sub>SS</sub>	30	-	300	µA
I <sub>OHi</sub>	transient pull-up current	HIGH during acknowledge (see Fig.14); V <sub>OH</sub> = V <sub>SS</sub> ; V <sub>DD</sub> = 2.5 V	-	-1	-	mA
C <sub>i</sub>	input capacitance		-	-	10	pF
C <sub>o</sub>	output capacitance		-	-	10	pF
<b>Port timing; C<sub>L</sub> ≤ 100 pF</b> (see Figs 10 and 11)						
t <sub>pv</sub>	output data valid		-	-	4	µs
t <sub>su</sub>	input data set-up time		0	-	-	µs
t <sub>h</sub>	input data hold time		4	-	-	µs
<b>Interrupt INT</b> (see Fig.13)						
I <sub>OL</sub>	LOW level output current	V <sub>OL</sub> = 0.4 V	1.6	-	-	mA
I <sub>L</sub>	leakage current	V <sub>I</sub> = V <sub>DD</sub> or V <sub>SS</sub>	-1	-	+1	µA
<b>TIMING; C<sub>L</sub> ≤ 100 pF</b>						
t <sub>iv</sub>	input data valid time		-	-	4	µs
t <sub>ir</sub>	reset delay time		-	-	4	µs
<b>Select inputs A0 to A2</b>						
V <sub>IL</sub>	LOW level input voltage		-0.5	-	+0.3V <sub>DD</sub>	V
V <sub>IH</sub>	HIGH level input voltage		0.7V <sub>DD</sub>	-	V <sub>DD</sub> + 0.5	V
I <sub>LI</sub>	input leakage current	pin at V <sub>DD</sub> or V <sub>SS</sub>	-250	-	+250	nA

**Note**

1. The Power-on reset circuit resets the I<sup>2</sup>C-bus logic with V<sub>DD</sub> < V<sub>POR</sub> and sets all I/Os to logic 1 (with current source to V<sub>DD</sub>).

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

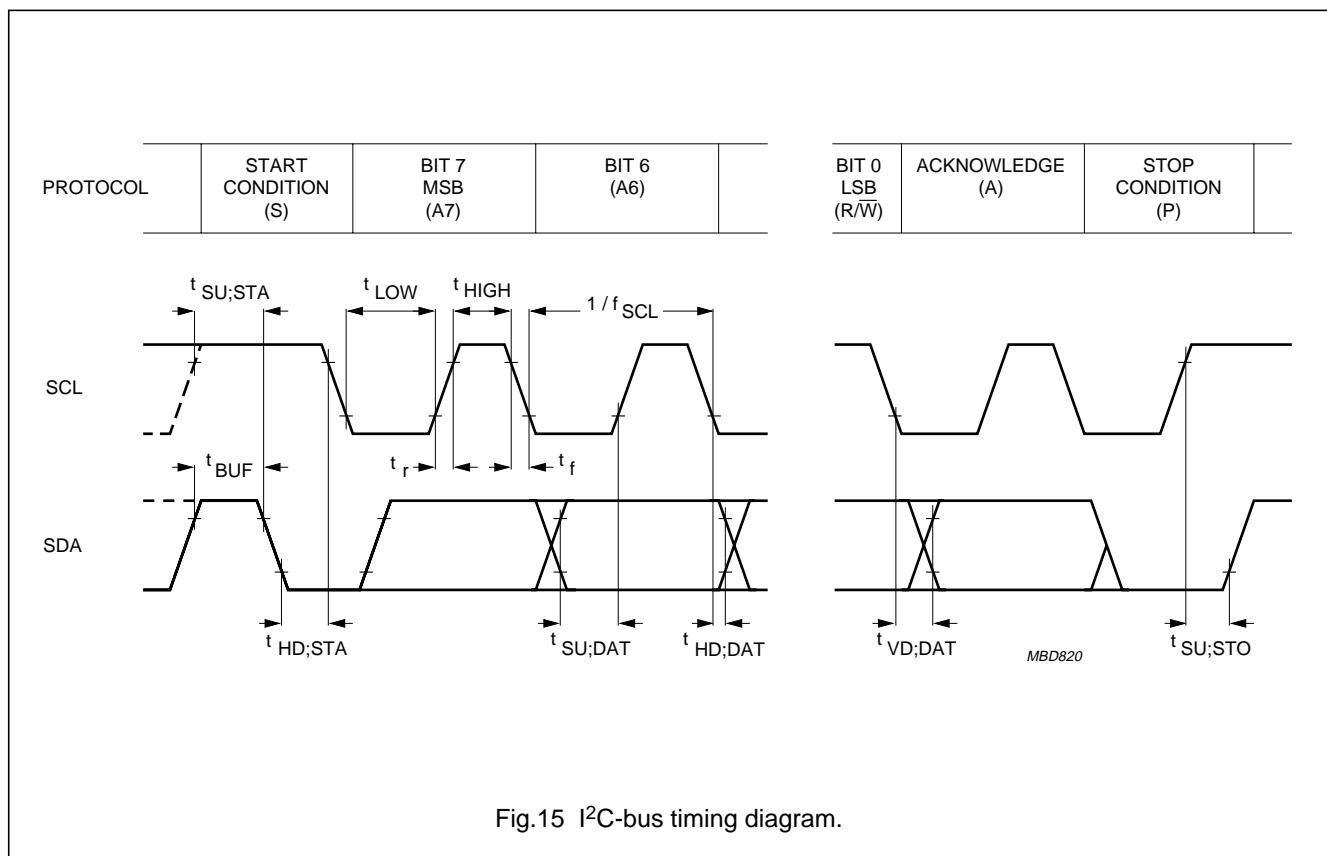
PCF8574

11 I<sup>2</sup>C-BUS TIMING CHARACTERISTICS

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
I <sup>2</sup> C-BUS TIMING (see Fig.15; note 1)					
$f_{SCL}$	SCL clock frequency	—	—	100	kHz
$t_{SW}$	tolerable spike width on bus	—	—	100	ns
$t_{BUF}$	bus free time	4.7	—	—	μs
$t_{SU;STA}$	START condition set-up time	4.7	—	—	μs
$t_{HD;STA}$	START condition hold time	4.0	—	—	μs
$t_{LOW}$	SCL LOW time	4.7	—	—	μs
$t_{HIGH}$	SCL HIGH time	4.0	—	—	μs
$t_r$	SCL and SDA rise time	—	—	1.0	μs
$t_f$	SCL and SDA fall time	—	—	0.3	μs
$t_{SU;DAT}$	data set-up time	250	—	—	ns
$t_{HD;DAT}$	data hold time	0	—	—	ns
$t_{VD;DAT}$	SCL LOW to data out valid	—	—	3.4	μs
$t_{SU;STO}$	STOP condition set-up time	4.0	—	—	μs

## Note

1. All the timing values are valid within the operating supply voltage and ambient temperature range and refer to  $V_{IL}$  and  $V_{IH}$  with an input voltage swing of  $V_{SS}$  to  $V_{DD}$ .



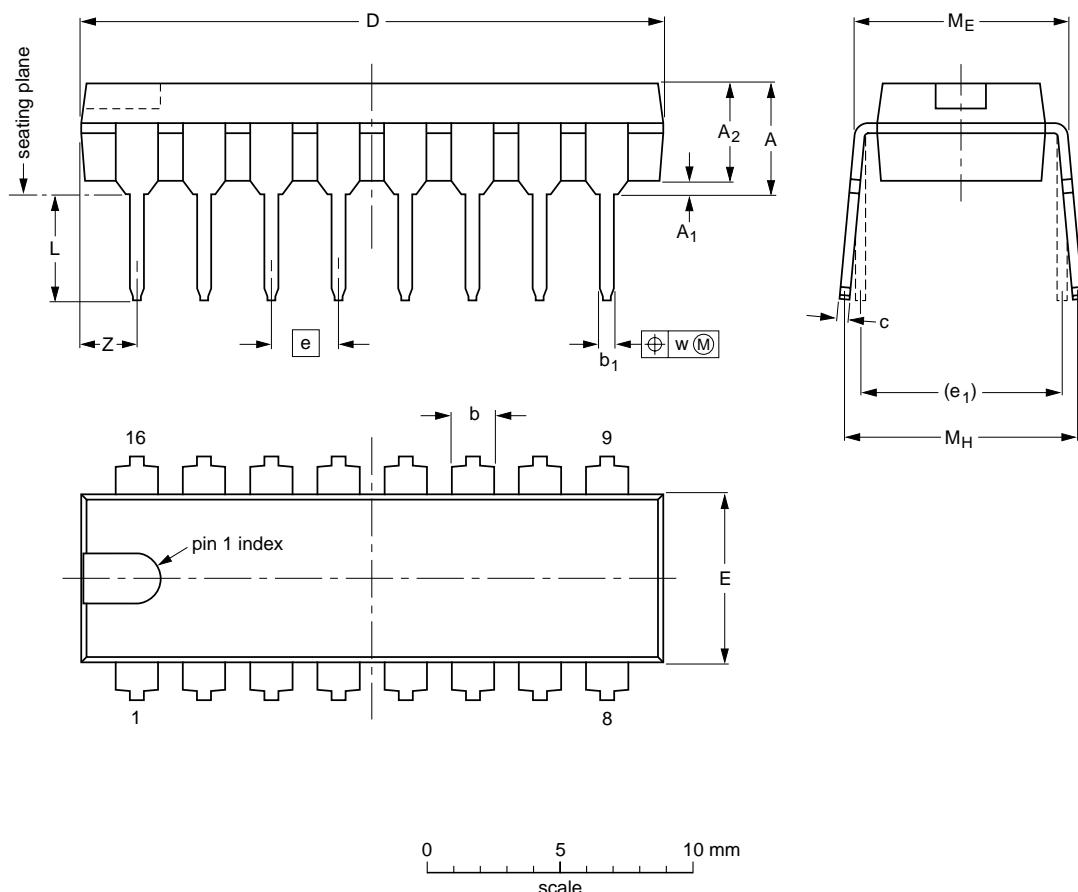
Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

## 12 PACKAGE OUTLINES

DIP16: plastic dual in-line package; 16 leads (300 mil); long body

SOT38-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A <sub>1</sub> min.	A <sub>2</sub> max.	b	b <sub>1</sub>	c	D <sup>(1)</sup>	E <sup>(1)</sup>	e	e <sub>1</sub>	L	M <sub>E</sub>	M <sub>H</sub>	w	Z <sup>(1)</sup> max.
mm	4.7	0.51	3.7	1.40 1.14	0.53 0.38	0.32 0.23	21.8 21.4	6.48 6.20	2.54	7.62	3.9 3.4	8.25 7.80	9.5 8.3	0.254	2.2
inches	0.19	0.020	0.15	0.055 0.045	0.021 0.015	0.013 0.009	0.86 0.84	0.26 0.24	0.10	0.30	0.15 0.13	0.32 0.31	0.37 0.33	0.01	0.087

## Note

- Plastic or metal protrusions of 0.25 mm maximum per side are not included.

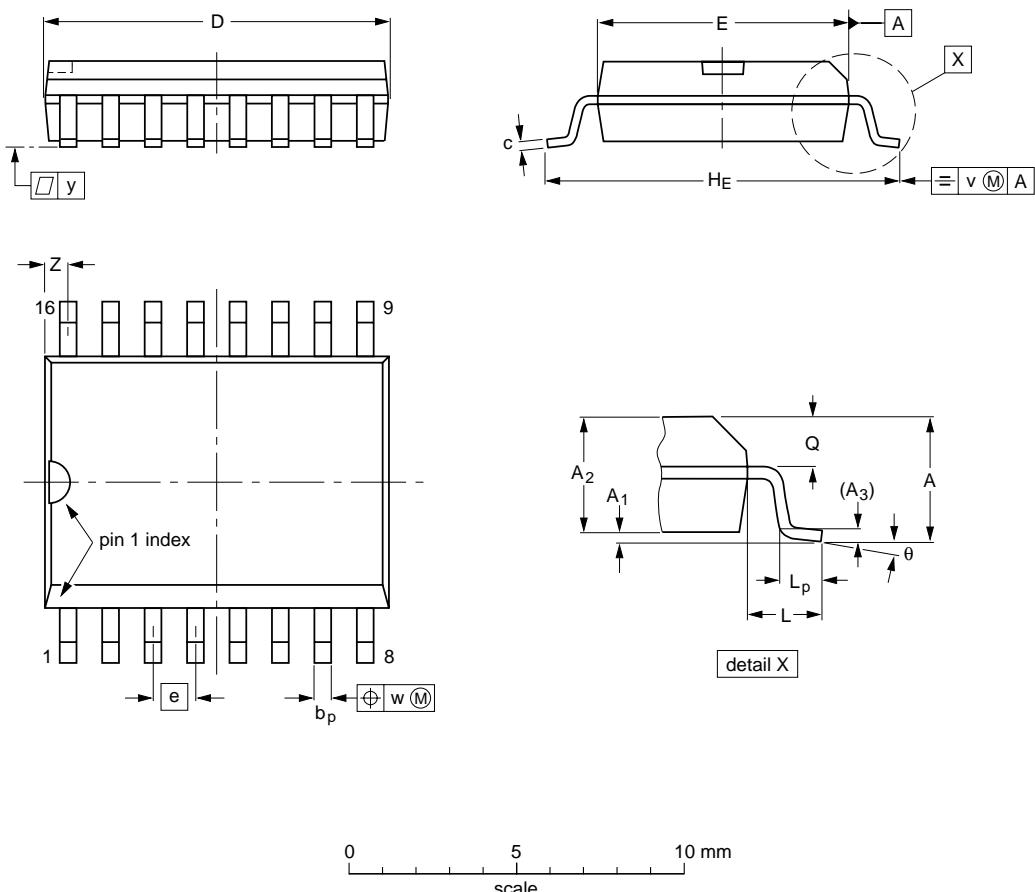
OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT38-1	050G09	MO-001AE				-92-10-02- 95-01-19

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

SO16: plastic small outline package; 16 leads; body width 7.5 mm

SOT162-1



## DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	b <sub>p</sub>	c	D <sup>(1)</sup>	E <sup>(1)</sup>	e	H <sub>E</sub>	L	L <sub>p</sub>	Q	v	w	y	z <sup>(1)</sup>	$\theta$
mm	2.65 0.10	0.30 2.25	2.45	0.25	0.49 0.36	0.32 0.23	10.5 10.1	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.1 1.0	0.25	0.25	0.1	0.9 0.4	8° 0°
inches	0.10 0.004	0.012 0.089	0.096	0.01	0.019 0.014	0.013 0.009	0.41 0.40	0.30 0.29	0.050	0.419 0.394	0.055	0.043 0.016	0.043 0.039	0.01	0.01	0.004	0.035 0.016	

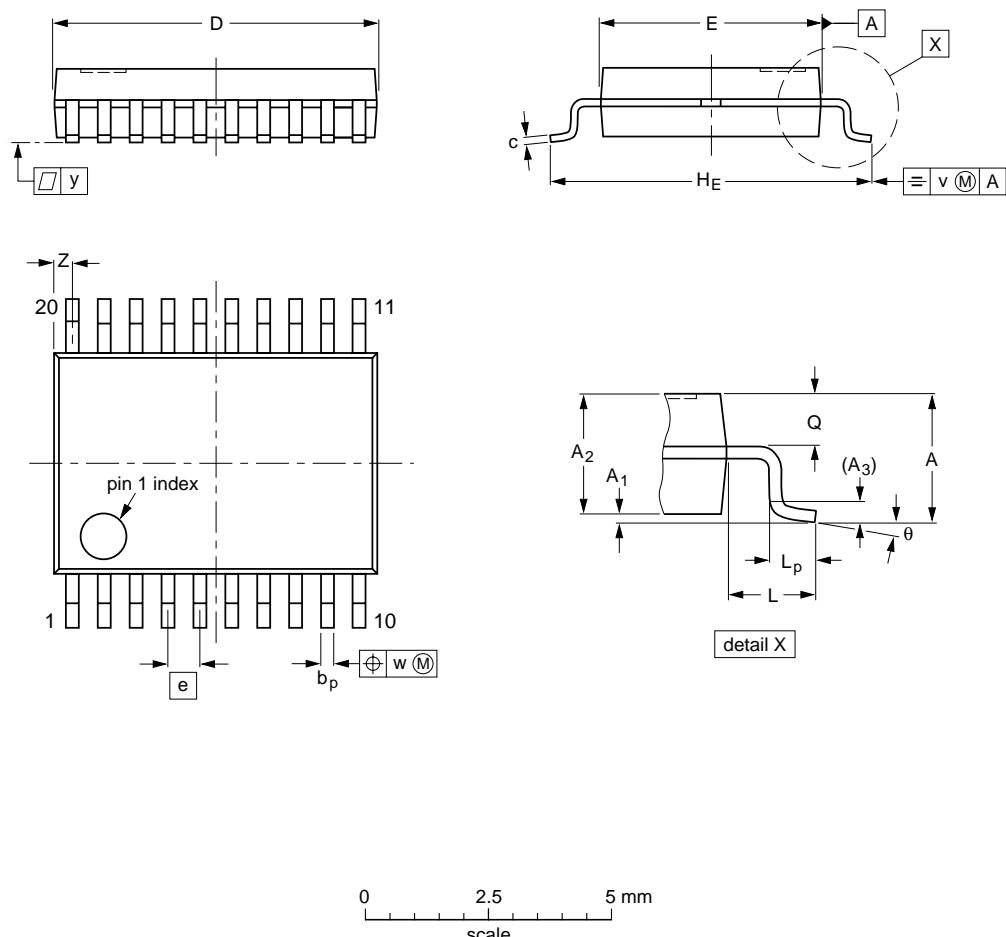
## Note

- Plastic or metal protrusions of 0.15 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT162-1	075E03	MS-013AA				-95-01-24 97-05-22

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

**SSOP20: plastic shrink small outline package; 20 leads; body width 4.4 mm****SOT266-1****DIMENSIONS (mm are the original dimensions)**

UNIT	A max.	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	b <sub>p</sub>	c	D <sup>(1)</sup>	E <sup>(1)</sup>	e	H <sub>E</sub>	L	L <sub>p</sub>	Q	v	w	y	Z <sup>(1)</sup>	θ
mm	1.5	0.15 0	1.4 1.2	0.25	0.32 0.20	0.20 0.13	6.6 6.4	4.5 4.3	0.65	6.6 6.2	1.0	0.75 0.45	0.65 0.45	0.2	0.13	0.1	0.48 0.18	10° 0°

**Note**

- Plastic or metal protrusions of 0.20 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT266-1						-90-04-05 95-02-25

# Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

## 13 SOLDERING

### 13.1 Introduction

There is no soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and surface mounted components are mixed on one printed-circuit board. However, wave soldering is not always suitable for surface mounted ICs, or for printed-circuits with high population densities. In these situations reflow soldering is often used.

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our "IC Package Databook" (order code 9398 652 90011).

### 13.2 DIP

#### 13.2.1 SOLDERING BY DIPPING OR BY WAVE

The maximum permissible temperature of the solder is 260 °C; solder at this temperature must not be in contact with the joint for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.

The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified maximum storage temperature ( $T_{stg\ max}$ ). If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

#### 13.2.2 REPAIRING SOLDERED JOINTS

Apply a low voltage soldering iron (less than 24 V) to the lead(s) of the package, below the seating plane or not more than 2 mm above it. If the temperature of the soldering iron bit is less than 300 °C it may remain in contact for up to 10 seconds. If the bit temperature is between 300 and 400 °C, contact may be up to 5 seconds.

### 13.3 SO and SSOP

#### 13.3.1 REFLOW SOLDERING

Reflow soldering techniques are suitable for all SO and SSOP packages.

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement.

Several techniques exist for reflowing; for example, thermal conduction by heated belt. Dwell times vary between 50 and 300 seconds depending on heating method. Typical reflow temperatures range from 215 to 250 °C.

Preheating is necessary to dry the paste and evaporate the binding agent. Preheating duration: 45 minutes at 45 °C.

#### 13.3.2 WAVE SOLDERING

Wave soldering is **not** recommended for SSOP packages. This is because of the likelihood of solder bridging due to closely-spaced leads and the possibility of incomplete solder penetration in multi-lead devices.

If wave soldering cannot be avoided, the following conditions must be observed:

- A double-wave (a turbulent wave with high upward pressure followed by a smooth laminar wave) soldering technique should be used.
- The longitudinal axis of the package footprint must be parallel to the solder flow and must incorporate solder thieves at the downstream end.

Even with these conditions, only consider wave soldering SSOP packages that have a body width of 4.4 mm, that is SSOP16 (SOT369-1) or SSOP20 (SOT266-1).

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Maximum permissible solder temperature is 260 °C, and maximum duration of package immersion in solder is 10 seconds, if cooled to less than 150 °C within 6 seconds. Typical dwell time is 4 seconds at 250 °C.

A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

#### 13.3.3 REPAIRING SOLDERED JOINTS

Fix the component by first soldering two diagonally-opposite end leads. Use only a low voltage soldering iron (less than 24 V) applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C. When using a dedicated tool, all other leads can be soldered in one operation within 2 to 5 seconds between 270 and 320 °C.

## Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

### 14 DEFINITIONS

<b>Data sheet status</b>	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
<b>Limiting values</b>	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
<b>Application information</b>	
Where application information is given, it is advisory and does not form part of the specification.	

### 15 LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

### 16 PURCHASE OF PHILIPS I<sup>2</sup>C COMPONENTS



Purchase of Philips I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C system provided the system conforms to the I<sup>2</sup>C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

---

**NOTES**

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

---

**NOTES**

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

---

**NOTES**

# **Philips Semiconductors – a worldwide company**

**Argentina:** see South America

**Australia:** 34 Waterloo Road, NORTH RYDE, NSW 2113,  
Tel. +61 2 9805 4455, Fax. +61 2 9805 4466

**Austria:** Computerstr. 6, A-1101 WIEN, P.O. Box 213,  
Tel. +43 1 60 101, Fax. +43 1 60 101 1210

**Belarus:** Hotel Minsk Business Center, Bld. 3, r. 1211, Volodarski Str. 6,  
220050 MINSK, Tel. +375 172 200 733, Fax. +375 172 200 773

**Belgium:** see The Netherlands

**Brazil:** see South America

**Bulgaria:** Philips Bulgaria Ltd., Energoproject, 15th floor,  
51 James Bourchier Blvd., 1407 SOFIA,  
Tel. +359 2 689 211, Fax. +359 2 689 102

**Canada:** PHILIPS SEMICONDUCTORS/COMPONENTS,  
Tel. +1 800 234 7381

**China/Hong Kong:** 501 Hong Kong Industrial Technology Centre,  
72 Tat Chee Avenue, Kowloon Tong, HONG KONG,  
Tel. +852 2319 7888, Fax. +852 2319 7700

**Colombia:** see South America

**Czech Republic:** see Austria

**Denmark:** Prags Boulevard 80, PB 1919, DK-2300 COPENHAGEN S,  
Tel. +45 32 88 2636, Fax. +45 31 57 1949

**Finland:** Sinikalliontie 3, FIN-02630 ESPOO,  
Tel. +358 9 615800, Fax. +358 9 61580/xxx

**France:** 4 Rue du Port-aux-Vins, BP317, 92156 SURESNES Cedex,  
Tel. +33 1 40 99 6161, Fax. +33 1 40 99 6427

**Germany:** Hammerbrookstraße 69, D-20097 HAMBURG,  
Tel. +49 40 23 53 60, Fax. +49 40 23 536 300

**Greece:** No. 15, 25th March Street, GR 17778 TAVROS/ATHENS,  
Tel. +30 1 4894 339/239, Fax. +30 1 4814 240

**Hungary:** see Austria

**India:** Philips INDIA Ltd, Shivsagar Estate, A Block, Dr. Annie Besant Rd.  
Worli, MUMBAI 400 018, Tel. +91 22 4938 541, Fax. +91 22 4938 722

**Indonesia:** see Singapore

**Ireland:** Newstead, Clonskeagh, DUBLIN 14,  
Tel. +353 1 7640 000, Fax. +353 1 7640 200

**Israel:** RAPAC Electronics, 7 Kehilat Saloniki St, TEL AVIV 61180,  
Tel. +972 3 645 0444, Fax. +972 3 649 1007

**Italy:** PHILIPS SEMICONDUCTORS, Piazza IV Novembre 3,  
20124 MILANO, Tel. +39 2 6752 2531, Fax. +39 2 6752 2557

**Japan:** Philips Bldg 13-37, Kohnan 2-chome, Minato-ku, TOKYO 108,  
Tel. +81 3 3740 5130, Fax. +81 3 3740 5077

**Korea:** Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL,  
Tel. +82 2 709 1412, Fax. +82 2 709 1415

**Malaysia:** No. 76 Jalan Universiti, 46200 PETALING JAYA, SELANGOR,  
Tel. +60 3 750 5214, Fax. +60 3 757 4880

**Mexico:** 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905,  
Tel. +9-5 800 234 7381

**Middle East:** see Italy

**Netherlands:** Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB,  
Tel. +31 40 27 82785, Fax. +31 40 27 88399

**New Zealand:** 2 Wagener Place, C.P.O. Box 1041, AUCKLAND,  
Tel. +64 9 849 4160, Fax. +64 9 849 7811

**Norway:** Box 1, Manglerud 0612, OSLO,  
Tel. +47 22 74 8000, Fax. +47 22 74 8341

**Philippines:** Philips Semiconductors Philippines Inc.,  
106 Valero St. Salcedo Village, P.O. Box 2108 MCC, MAKATI,  
Metro MANILA, Tel. +63 2 816 6380, Fax. +63 2 817 3474

**Poland:** Ul. Lukiska 10, PL 04-123 WARSZAWA,  
Tel. +48 22 612 2831, Fax. +48 22 612 2327

**Portugal:** see Spain

**Romania:** see Italy

**Russia:** Philips Russia, Ul. Usatcheva 35A, 119048 MOSCOW,  
Tel. +7 095 755 6918, Fax. +7 095 755 6919

**Singapore:** Lorong 1, Toa Payoh, SINGAPORE 1231,  
Tel. +65 350 2538, Fax. +65 251 6500

**Slovakia:** see Austria

**Slovenia:** see Italy

**South Africa:** S.A. PHILIPS Pty Ltd., 195-215 Main Road Martindale,  
2092 JOHANNESBURG, P.O. Box 7430 Johannesburg 2000,  
Tel. +27 11 470 5911, Fax. +27 11 470 5494

**South America:** Rua do Rocio 220, 5th floor, Suite 51,  
04552-903 São Paulo, SÃO PAULO - SP, Brazil,  
Tel. +55 11 821 2333, Fax. +55 11 829 1849

**Spain:** Balmes 22, 08007 BARCELONA,  
Tel. +34 3 301 6312, Fax. +34 3 301 4107

**Sweden:** Kottbygatan 7, Akalla, S-16485 STOCKHOLM,  
Tel. +46 8 632 2000, Fax. +46 8 632 2745

**Switzerland:** Allmendstrasse 140, CH-8027 ZÜRICH,  
Tel. +41 1 488 2686, Fax. +41 1 481 7730

**Taiwan:** Philips Semiconductors, 6F, No. 96, Chien Kuo N. Rd., Sec. 1,  
TAIPEI, Taiwan Tel. +886 2 2134 2870, Fax. +886 2 2134 2874

**Thailand:** PHILIPS ELECTRONICS (THAILAND) Ltd.,  
209/2 Sanpavut-Bangna Road Prakanong, BANGKOK 10260,  
Tel. +66 2 745 4090, Fax. +66 2 398 0793

**Turkey:** Talatpasa Cad. No. 5, 80640 GÜLTEPE/İSTANBUL,  
Tel. +90 212 279 2770, Fax. +90 212 282 6707

**Ukraine:** PHILIPS UKRAINE, 4 Patrice Lumumba str., Building B, Floor 7,  
252042 KIEV, Tel. +380 44 264 2776, Fax. +380 44 268 0461

**United Kingdom:** Philips Semiconductors Ltd., 276 Bath Road, Hayes,  
MIDDLESEX UB3 5BX, Tel. +44 181 730 5000, Fax. +44 181 754 8421

**United States:** 811 East Arques Avenue, SUNNYVALE, CA 94088-3409,  
Tel. +1 800 234 7381

**Uruguay:** see South America

**Vietnam:** see Singapore

**Yugoslavia:** PHILIPS, Trg N. Pasica 5/v, 11000 BEOGRAD,  
Tel. +381 11 625 344, Fax. +381 11 635 777

**For all other countries apply to:** Philips Semiconductors, Marketing & Sales Communications,  
Building BE-p, P.O. Box 218, 5600 MD EINDHOVEN, The Netherlands, Fax. +31 40 27 24825

**Internet:** <http://www.semiconductors.philips.com>

© Philips Electronics N.V. 1997

SCA53

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Printed in The Netherlands

417067/1200/02/PP24

Date of release: 1997 Apr 02

Document order number: 9397 750 01758

*Let's make things better.*

**Philips**  
**Semiconductors**



**PHILIPS**

## SG90 Micro Servo

Servo motor for Arduino and Raspberry-Pi robotics projects.

Typical use: Model aircraft, cars and robots.

Rotates forward or backwards to a given position.

Bi-directional rotation - pulse duration determines the direction.

Item no: 87897 single pack, 90720 pack of 4 pcs.

Model no: SG90

Weight: 11g.

Rotation Angle: 180°

Torque: 1,8 kg/cm (11 Ncm) (at 4,8 V).

Torque: 2,4 kg/cm (15 Ncm) (at 6 V).

Speed: 0,12 sec/60° (at 4,8 V).

Speed: 0,10 sec/60° (at 6 V).

Linear response to PWM for easy ramping.

Operating voltage: 4,8-6 V.

Operating temperature: -10 to 50 °C.

Current max: < 600 mA.

Cable length: 250 mm.

Connector type: JR / Futaba / GWS

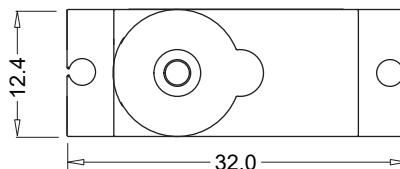
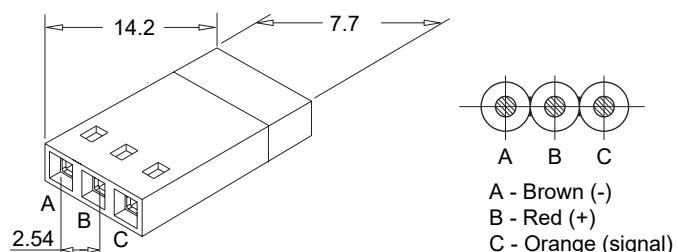
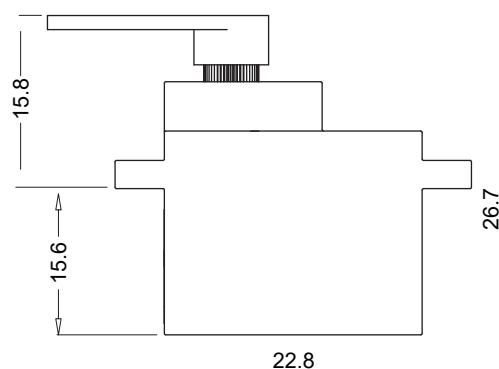
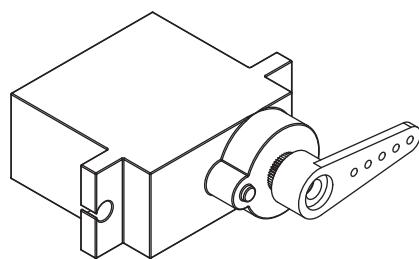
Breadboard friendly connector 2,54 mm pitch.

Connector wire gauge: 28 AWG.

Control system: PWM (Pulse Width Modulation)

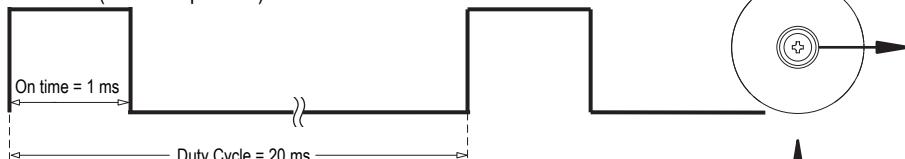
Pulse Frequency / Duty cycle: 50 Hz / 20 ms square wave

Direction w/ Increasing PWM Signal: Counter Clockwise

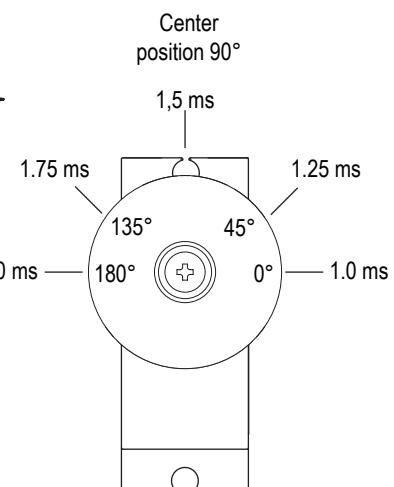


### 180° servo PWM signal timings

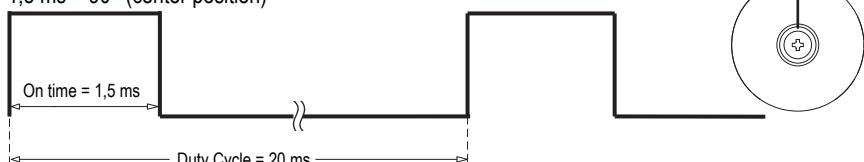
1 ms = 0° (minimum position)



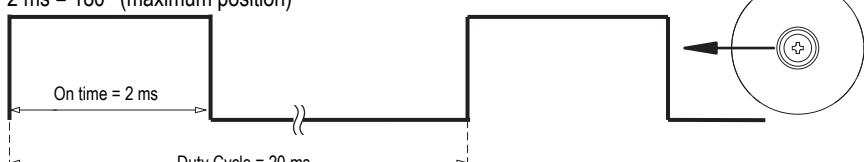
Center position 90°



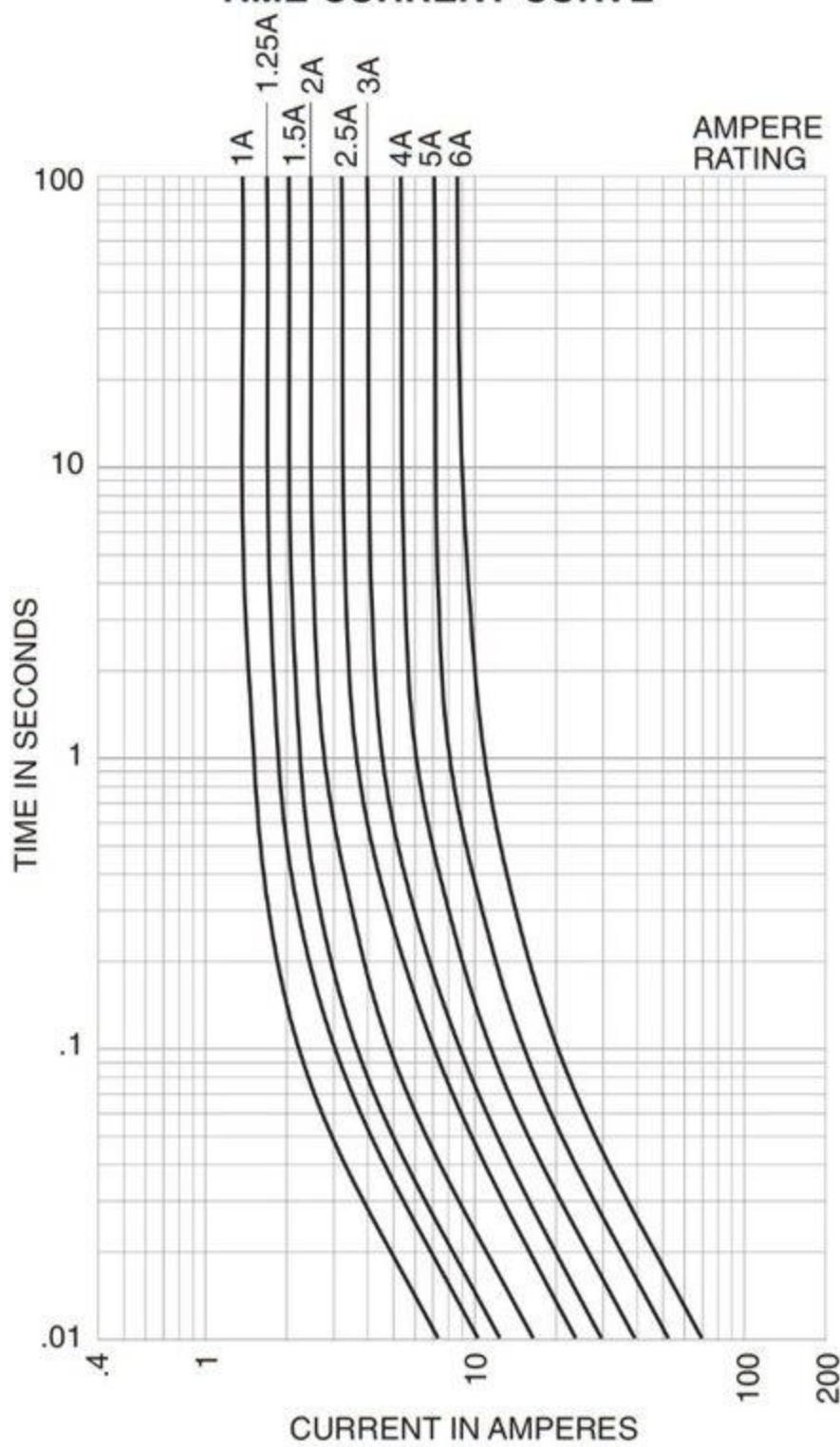
1,5 ms = 90° (center position)



2 ms = 180° (maximum position)



## TIME CURRENT CURVE



# Basys 3 FPGA Manual

Elizabethtown College

Clay Buxton

v1.0 – Clay Buxton 5/12/19  
v0.2 – Clay Buxton 4/26/19  
v0.1 – Clay Buxton 3/25/19

This manual is a work in progress. Sections with no prefix are finished and complete. Sections marked with a [WIP] mean they are currently work in progress, these may not be finished. Segments marked [NS] mean they have not been started and have yet to be started.

This manual in no way, shape, or form should be taken as straight truth. I've been learning as well as writing this manual, it is prone to mistakes, and many beginner errors. If other resources conflict with what this manual says, take the information told in other sources.

## Table of Contents:

- [Introduction to FPGA's](#)
- [Creating Your First Project and Basic Vivado UI](#)
- [Inputs, Outputs and Configuring Design Constraints](#)
- [Basic Verilog](#)
- [Programming the FPGA](#)
- [Test Benches and Timing Diagrams](#)
- [PMod Interfacing](#)
- [Appendix](#)

## Resources:

### **Digilent's Basys 3 Github Repository** <https://github.com/Digilent/Basys3>

This repository holds the constraints file for the Basys 3 as well as a few helpful example projects. It is an excellent resource for looking at some examples in Verilog for the board.

### **Clays EGR333 Github Repository** <https://github.com/clbx/EGR333>

This repository holds all the code shown in this manual and some simple projects to help learn Verilog

### **MIT Intro to Verilog** [http://web.mit.edu/6.111/www/f2016/handouts/L03\\_4.pdf](http://web.mit.edu/6.111/www/f2016/handouts/L03_4.pdf)

These slides are a good, brief, and complete look into the basics of Verilog. Very Useful

# 1 Introduction to FPGA's

This section does not hold any information on how to program or operate the boards. Operable information starts in section 2

FPGA's or Field Programmable Gate Arrays are development boards that house a chip that can be programmed to perform as a set of logic blocks functionally.

This manual is strictly for the Basys 3 housing the Artix 7 chip. Vivado is used to program this chip, and any reference to programming should be assumed to be done through Vivado unless otherwise specified.

## FPGA vs. Breadboarding vs. Logisim

- FPGA's allow for a wide range of features that designing a circuit manually on a breadboard and simulating one like in Logisim do not provide.
- Using Vivado, we can streamline the design, testing, and implementation of a circuit. Unlike with breadboarding it must all be done manually and Logisim where it is challenging to implement.
- FPGA's allow for quick re-use. An FPGA has the ability to be reprogrammed in a matter of minutes. Logisim also provides this, however when hand-building circuits this becomes time-consuming
- Designing a circuit in Vivado allows for the use of HDL and other programming languages. This gives the ability to rapidly develop designs
- Voltage, current, and power are irrelevant when planning for an FPGA; all this is taken care of behind the scenes.

Designing for an FPGA allows for much higher density designs. An FPGA is able to support more circuitry than 100 breadboards and models in Logisim face the technical limitations of the software along with a lack of an efficient way to manage many designs.

FPGAs are made from a single component, a logic cell. These logic cells can be reconfigured and re-assigned as needed by the program being written to them. Logic cells contain three main parts, a look-up table or LUT, a D flip-flop, and a multiplexer. A LUT can function any logic function based on the number of pins they are allocated. They can be assigned to any logic input and output and are not limited to gates or simple components. This design allows LUTs to be used together to implement any logic and are used to implement logic on an FPGA.

The Basys 3 boards are programming using the Vivado Software Suite. Vivado is the replacement for the old Xilinx ISE design suite from 2014 onwards. Vivado does not support any older chips, and Xilinx ISE does not support any newer chips. Unlike ISE which relied primarily on gate design, Vivado utilizes mostly the use of HDL and supports designs built with high-level languages like C and C++.

# 2 Setting Up Your First Project and Basic Vivado UI

Vivado is the design suite used to write programs onto the FPGAs. At first, it can be a very intimidating program. In the first few sections of this manual, the FPGAs are programmed using a **Hardware Descriptive Language (HDL)**. Hardware Descriptive languages are different from most high-level languages like C and Java. Instead of writing code traditionally, HDL's describe hardware components. This manual uses Verilog for all examples and tutorials in the first few sections. However, this manual only covers simple Verilog (enough to get a simple circuit up and running on the FPGAs) in depth topics are not included as there are many better resources to use for that.

To start Vivado find the shortcut on the desktop or find it in the start menu. Make sure you are launching Vivado and not Vivado HLS. The shortcut should be something along the lines of **Vivado 20XX.Y** where XX is the year and Y the version number



Once Vivado has loaded you should see a Project Screen with three sections **Quick Start**, **Tasks**, and **Learning Center**, along with the recent projects window. Click **Create Project** to start with a new project.

A screenshot of the 'Create Project' dialog box. It shows a list of project types: 'RTL Project' (selected), 'Post-synthesis Project', 'J/I Planning Project', 'Imported Project', and 'Example Project'. Under 'Do not specify sources at this time', the 'Do not specify sources at this time' checkbox is checked. Below the dialog is a 'Parts' search interface with filters for Category (All), Family (All), Package (All), Speed (All), Temperature (All), and a search bar for 'xc7a35tcmpg236-1'.

[1] The first window presents you with the name and location of the project. Where you save this doesn't matter, put it wherever is most convenient for you. I did not notice any reduced performance by having it in my public/private folders.

[2] The second window asks you to specify the **Project Type**. Select **RTL Project** and be sure that "**Do not specify sources at this time**" is **checked**. The box should be because you do not currently have any sources to add and want to open a blank project. RTL project is the standard project in Vivado that targets the FPGAs.

[3] The third window asks you to specify the board that we are using. The part number for the Basys 3 FPGAs is **XC7A35TCPG236-1**. Select it and click next, then finish.

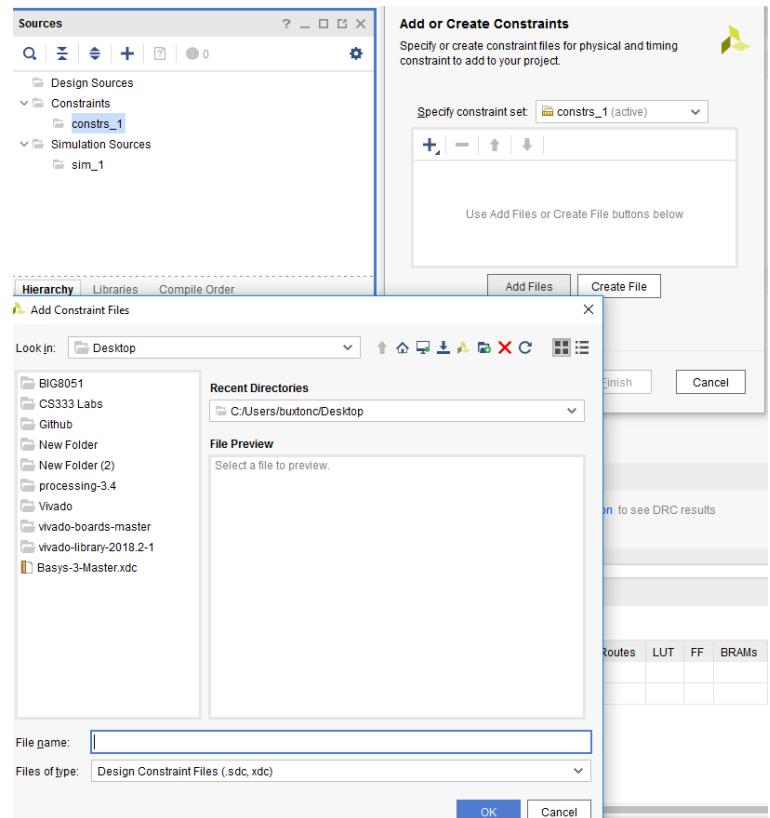
You should now see the main screen for the Vivado Development Environment. There are three main sections here. You can see **Flow Navigator** to the left, **Project Manager** in the middle, and the **Console and other outputs** on the bottom of the Project Manager.

In the Project manager, there are three smaller boxes. The **Sources**, **Project Summary**, and **Properties**. You can ignore the properties and summary boxes for now

In the **Sources** box, you'll see three folders, **Design Sources**, **Constraints**, and **Simulation Sources**. Our first focus is going to be on the **Constraints**. A **Constraint** is a file that tells Vivado what hardware to use and where on the board it is. The constraint files have a .xdc file extension. This file is provided to you, though you may write your own. It is called "**Basys-3-Master.xdc**" or something along those lines. Click the "plus" icon at the top of the sources window. The next section describes how to configure the design constraints for the FPGA.

The **Design Sources** folder is where you put the source code for the project. That process is covered in a later step.

Now connect the MicroUSB cable to the FPGA and your Computer and turn the "on" switch on the FPGA. The FPGA should power on with the example program running.



# Summary

To set up a project, follow the steps:

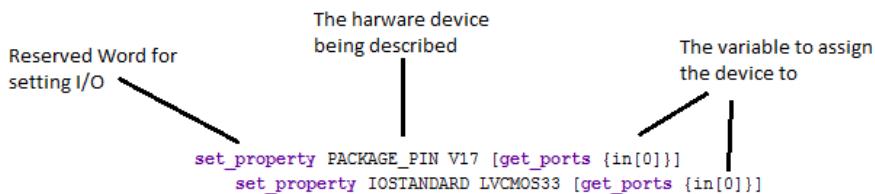
1. Set the location of the project and give it a name
2. Select **RTL Project**, make check **Do not specify design sources**
3. Select the **XC7A35TCPG236-1**
4. Import the **Basys-3-Master.xdc** design constraint

# 3 Inputs, Outputs and configuring

## Design Constraints

The Basys 3 Constraints file can be found in the Digilent Basys 3 Github Repository

To configure what inputs and outputs you are using in your project, and assign the hardware to a variable in software, you need to edit the constraints file. When you open the file **Constraints/constrs\_1/Basys-3-Master.xdc** it will open long file with many lines starting with **set\_property**. Every two lines of this file describes one input or output on the FPGA. Below is a diagram which describes what each of the parts of the lines do.



**set\_property** Is a reserved word that tells Vivado that we are setting the property of a hardware device.

**PACKAGE\_PIN** Describes which device on the FPGA we are accessing. Here it is referencing V17 which is the first switch. Sometimes the pin on the board can be hard to identify it is generally the number in parentheses printed on the board near the device you want to use.

**get\_ports** Assigns the hardware device to a specific variable. Here it is putting the switch to the first position of the “in” array.

For example, if we wanted to set the first LED on the board to the variable `led` in code. We would put the following lines in the design constraints file

```
set_property PACKAGE_PIN V16 [get_ports {led}]
set_property IOSTANDARD LVCMOS33 [get_ports {led}]
```

The **IOSTANDARD** portion of the line tells Vivado what protocol is used to talk to the hardware though you don’t need to worry about this, it won’t change.

To correctly setup which devices you are using, comment and uncomment the lines of the things you need to use. Make sure to keep unused items commented and uncomment all used items. If you don’t use a device described in the constraints, your project will not build. Similarly, if you try and use commented device, it will not work, and you’ll get similar results.

There are 12 sections in this file each correlates to a different type of input or output on the board.

### IMPORTANT

Before you’re done editing your constraints file make sure these things are done:

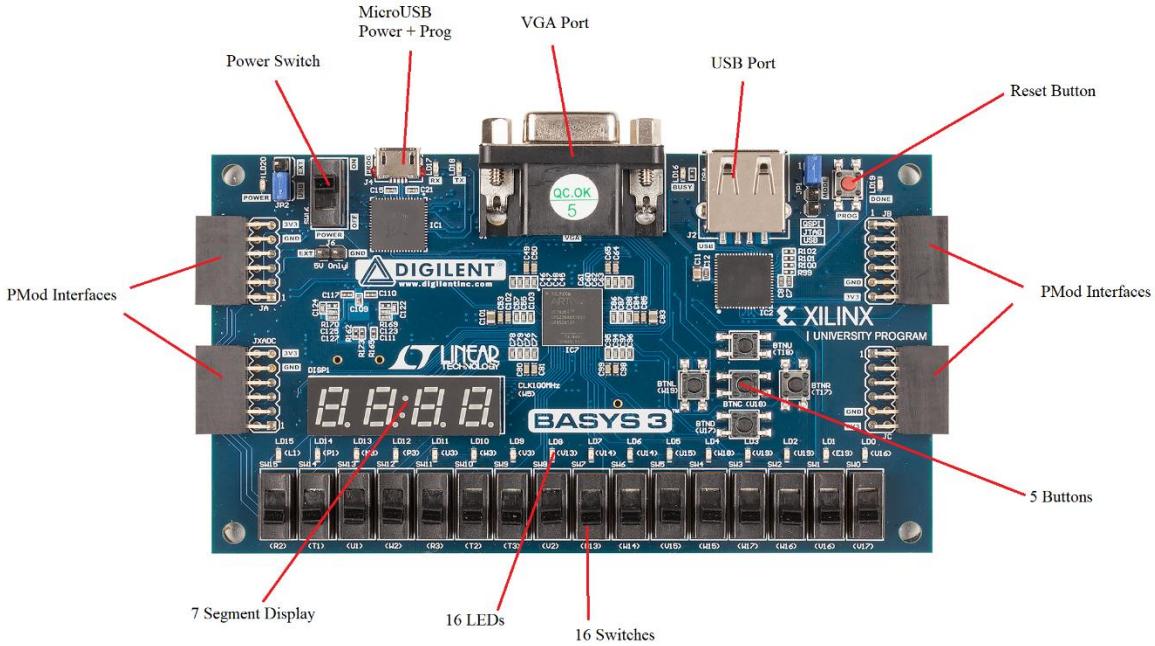
[1] All the variables that are assigned in the constraints are used in your code

[2] You don’t use a variable in the code that is undefined in the constraints

Otherwise your program will not compile!

The main ones that are covered in the early sections are the **switches**, **LEDs**, **clock**, and **buttons**.

Some of the inputs are assigned directly; others require different ways of communicating with the hardware. Things like the switches and LEDs are directly referenced, but some of the more complicated devices cannot. (ex: `led[1] = true` will turn on a light) however most of the other devices have specific ways to communicate with the board; these are covered in their respective sections.



## List of Inputs/Outputs:

**Clock:** The first section is for the internal clock that Verilog projects can use. It has an extra line that allows for setting clock parameters.

**Switches:** This section is for the 16 flip switches along the bottom of the board. You should use these in an array. These switches are referenced directly.

**LEDs:** The 16 Small LEDs above the switches along the bottom of the board. It is recommended to use these in an array. There are other referenceable LEDs on the board, but they are used for other functions, and you should stick to the 16 along the bottom. These can be referenced directly.

**7 Segment Display:** A 4 digit seven segment display.

**Buttons:** There are five referencable buttons in the middle of the board. The buttons are **not** debounced and can be referenced directly

**PMod Headers:** These are the 4, 12 pin connectors off the sides of the board. They can be used to communicate with external things. These are covered extensively in another section

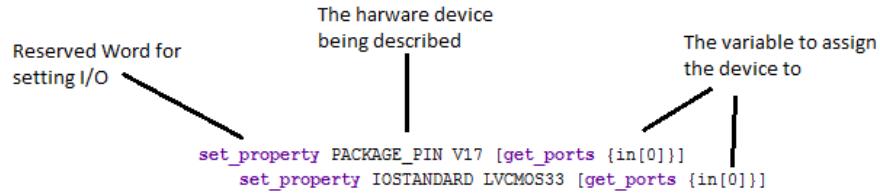
**VGA Connector:** VGA Connector to allow to display graphics. This is not covered in the current version of the manual

**USB Connections:** Way to communicate over USB. This is not covered in the current version of the manual

**Quad SPI Flash:** Flash memory. This is not covered in the current version of the manual

# Summary

The constraints file holds the information of what is being used and what isn't being used for the project. In each set of lines, you can set what hardware is being used and assign it to certain variables by referencing this chart.



# 4 Basic Verilog (4 Bit Adder)

As previously mentioned Verilog is different than a high-level language like C or Java. Verilog describes hardware like wires, gates and other hardware components. Things like `wire`, `xor`, and `reg` are reserved words instead of `int`, `double`, and `char`. They describe physical components that make up a circuit. Verilog still has things like logic and loops, which are used in the circuits.

The absolute best thing to do when starting to program in Verilog is to remember you're not writing a program, you're writing about a circuit.

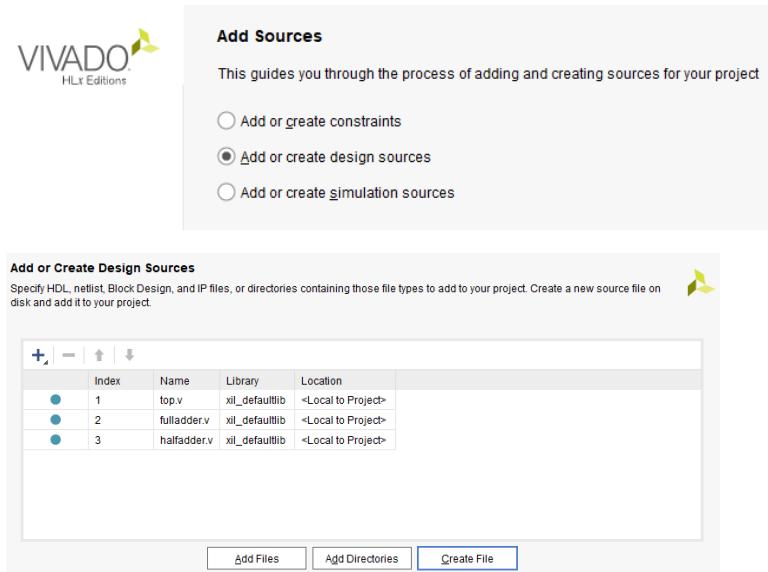
By the end of this section, you'll have a full 4-bit adder in Verilog and program it to the FPGA in the next.

The first thing we want to do is create our source files. Since we are going to create a 4-bit full adder, we are going to make 3 circuits; a half-adder, a full-adder, and then our assembled 4-bit full adder. Each file is a module, and a module is essentially a sub-circuit from Logisim. A circuit that is defined in one place and can be used elsewhere. To assemble our full adder, we need 3 files.

To create source files click the “**plus**” icon in the **Sources** panel. Select **Add or Create Design Sources** and next.

In the next panel, it shows all of the files that are being added to the project. Since we don't have any files to import, we want to create the files. Click **Create File**, and call the first file **top**. The top is similar to main for an HLL.

Make sure the file type is **Verilog**, and the file location is **<Local to Project>**. Then create **fulladder** and **halfadder** with the same settings.



Once you click finish, a new window will appear that allows you to declare what the inputs and outputs are of each of the modules. All this does is autogenerated the code inputs and outputs, you usually can fill this out, but for this manual, we're going to do it manually. Leave it blank and hit **OK** and **Yes** on the next window.

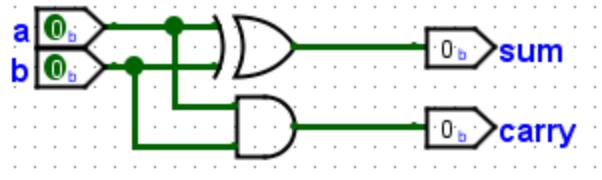


You should now see the 3 files in the Sources folder in the sources pane. You'll notice that one of them is bolded, this is what Vivado has currently selected as the **top** if “top.v” is not currently selected, right click on it and select **set as top**.

Now that we have our source files laid out, we can begin to write our code.

## Half Adder

At this point, you should know how to design a half adder in something like Logisim. It takes in 2 inputs, “adds” them using an XOR and AND gate and the outputs them on an output and carry outputs.



When opening the file, you’ll see some code already generated.

```
'timescale 1ns / 1ps
```

Sets the timescale for the project, leave this as it is, it’s not important right now

The large commented block is to put in information about the file into it. It is important to always fill this out this for every file, no matter how mundane it might feel. There is an example given in the appendix.

```
halfadder.v
module halfadder(
);
endmodule
```

Shown to the left is the base of the module. As you can see it looks very similar to a class in an HLL. It works similarly. Inside the parentheses, we’ll define what the inputs and outputs of the circuit are

Let’s write some code!

```
halfadder.v
module halfadder(
    input a, b,
    output sum, cout
);
endmodule
```

Inside the parentheses, we put **a**, **b**, **sum**, and **cout**. You also assign inputs and outputs here as shown.

Order here matters, when you call the circuit in another design, you’ll give the variables in the order described here

Using gates in Verilog is very similar to calling a function in another language. There are functions for XOR, AND, OR, XOR, etc. To use these you give them arguments just like you would a function in Java or C. The first parameter is the output of the gate, and then the following are the input. You can give it as many inputs as you please.

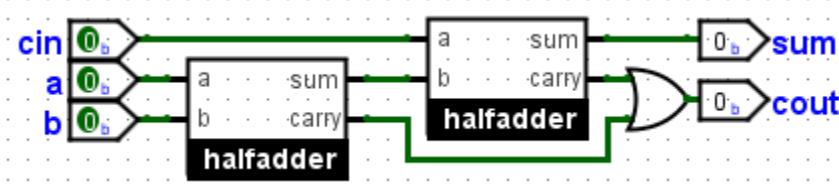
```
halfadder.v
module halfadder(
    input a, b,
    output sum, cout
);

    xor(sum,a,b);
    and(cout,a,b);
endmodule
```

Here we describe an AND gate and an XOR gate. The AND gate takes in **a** and **b** and outputs to **cout**. The XOR gate also takes in **a** and **b** but outputs to **sum**, just like in the Logisim circuit above

To the left is a completed half adder. Now we can move on to the next part, the full adder.

## Full Adder



As you know a full adder uses two half adders to add 3, 1-bit numbers and output the sum

fulladder.v

```
module fulladder(
    input a,b,cin,
    output sum,cout
);

endmodule
```

Go to your **fulladder.v** to start work on the full adder module.

Like we did for the half adder, assign your inputs and outputs.

Using another module is very similar to calling a gate. Make sure to give the module the inputs and outputs in the same order you defined them in code

```
fulladder.v

module fulladder(
    input a,b,cin,
    output sum,cout
);

    wire sum2sum, cout0, cout1;

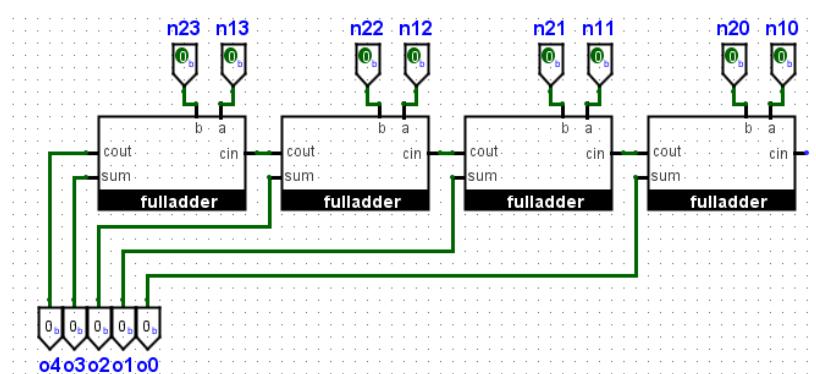
    halfadder(a,b,sum2sum,cout0);
    halfadder(sum2sum,(cin,sum),sum,cout1);
    or(cout,cout0,cout1);

endmodule
```

Here we had to add 3 wires as well; wires connect different parts of your circuit. We didn't need them before because inputs and outputs are typed as a wire, but now since we have wires in the internals of our circuits, we need to define them to connect the half-adders.

## The 4 Bit Adder

Now that we have all of the components for our 4-bit adder it's time to assemble it. We just put 4 full adders in a row and connect them. In the logisim design, I had to shrink down the output and input names. n10 refers to num1[0] in the code, n20 refers to num2[0], n11 refers to num1[1], and so on.



```
top.v ━━━━━━
module top(
    input [3:0] num1,
    input [3:0] num2,
    output [4:0] out,
);
endmodule
```

You can see here that some of the inputs look a little bit different than before. The `[3:0]` in front of the variables mean they are **busses**. Busses are wires that can handle more than one bit at a time and are similar to arrays from an HLL. A 3:0 bus holds 4 bits, from bit 0 to bit 3. Busses do not have a width limit (within reason).

```
top.v ━━━━━━
module top(
    input [3:0] num1,
    input [3:0] num2,
    output [4:0] out,
);
wire cout0,cout1,cout2;
endmodule
```

We'll need to create some wires so we can connect our full adders.

```
top.v ━━━━━━
module top(
    input [3:0] num1,
    input [3:0] num2,
    output [4:0] out
);
wire cout0,cout1,cout2;

fulladder(1'b0,num1[0],num2[0],out[0],cout0);
fulladder(cout0,num1[1],num2[1],out[1],cout1);
fulladder(cout1,num1[2],num2[2],out[2],cout2);
fulladder(cout2,num1[3],num2[3],out[3],out[4]);

endmodule
```

Now we can add in our four full adders to the circuit. There's two new things in this snippet too `num1[0]` a bus reference and `1'b0` a literal.

Busses are referenced like arrays in an HLL. Just give the name and index as you would regularly

Constants are defined in either binary, octal, decimal or hexadecimal. The very first number defines how many bits the number is. In this case, we just wanted a one bit zero to fill our first full adders carry in. The `'b` tells it that the number we're giving it is in binary, you can give it `'o`, `'d`, or `'h` for octal, decimal, and hexadecimal respectively

Our full adder is now complete! In the next section, we'll program the board.

# 5 Programming the FPGA

Armed with a 4-bit adder and a general idea of how constraints work, we can now edit our constraints file to fit the needs of our program. In the constraints, we need to set inputs and outputs for the top file.

We have a 2, 4-bit inputs and then a 5-bit output. Using switches for the inputs seems to make the most sense so we'll set our first four switches to **num1[]** and the next four switches to **num2[]**

Basys-3-Master.xdc

```
## Switches
set_property PACKAGE_PIN V17 [get_ports {num1[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {num1[0]}]
set_property PACKAGE_PIN V16 [get_ports {num1[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {num1[1]}]
set_property PACKAGE_PIN W16 [get_ports {num1[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {num1[2]}]
set_property PACKAGE_PIN W17 [get_ports {num1[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {num1[3]}]
```

Open your **Basys-3-Master.xdc** and assign the first four switches to num1[0] through num1[3]

Basys-3-Master.xdc

```
set_property PACKAGE_PIN W15 [get_ports {num2[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {num2[0]}]
set_property PACKAGE_PIN V15 [get_ports {num2[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {num2[1]}]
set_property PACKAGE_PIN W14 [get_ports {num2[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {num2[2]}]
set_property PACKAGE_PIN W13 [get_ports {num2[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {num2[3]}]
```

Then do the same for the next four switches.

Basys-3-Master.xdc

```
## LEDs
set_property PACKAGE_PIN U16 [get_ports {out[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {out[0]}]
set_property PACKAGE_PIN E19 [get_ports {out[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {out[1]}]
set_property PACKAGE_PIN U19 [get_ports {out[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {out[2]}]
set_property PACKAGE_PIN V19 [get_ports {out[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {out[3]}]
set_property PACKAGE_PIN W18 [get_ports {out[4]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {out[4]}]
```

And then set the outputs

Make sure unused lines are commented out; otherwise your project will not build!

Now it's time to put the project onto the board. Make sure your FPGA is plugged in and powered on, you should see the demo program running. In the **Project Manager** window towards the bottom you'll see a button called **Generate Bitstream** this runs all the necessary steps to compile your project.

There are three steps from Verilog to board.

The first is **Synthesis**. The Synthesis synthesizes all of your code into a gate level design. Your Verilog code turns into gates as it would on a physical circuit

The second is **Implementation**. During implementation, Vivado optimizes the gate design to run faster and use less power. If you were to take a look at the circuit after optimization, it would look unrecognizable since the circuit has been re-arranged to make it run better on the board.

The last step is to **Generate Bitstream**; this puts your gate design into a compiled binary file that the FPGA knows how to read. At this point, the design is no longer readable by anything except the board itself.

You can run all of these steps individually but starting any of the later ones will start the earlier ones if changes were made.

At this point run **Generate Bitstream**, be patient as this can take a few minutes. You can see what step it is currently on in the top right of the window

Once this is finished Click **Open Hardware Manager** to open the hardware manager.

In the green ribbon at the top of the window, you'll see a message saying **No hardware target is open**. Click **Open Target** and select **Auto Connect**, make sure the FPGA is on.

How the FPGA is connected and in the same ribbon you should see a **Program Device** button, click this. Press **Program** on the next window

The FPGA is now running your 4 Bit Adder!

# 6Verilog Snippets

This section isn't a tutorial as much as snippets and facts about Verilog I've learned going along. These are my observations and could be incorrect.

**Using always():** In many tutorials, you'll see `always(*) begin end` from my experience Vivado/the FPGA's do not like this. The item inside the parentheses in an always statement is when the loop is triggered. If you did `always(button) begin end` It would enter that whenever a button was high. However, Vivado will give you an error if you use anything besides a clock. So to my knowledge clk is the only thing that is valid in there without Vivado getting upset.

I don't believe this is right though since many tutorials use inputs other than clocks in their code. I'm not sure why Vivado doesn't like this.

## Difference between = and <= :

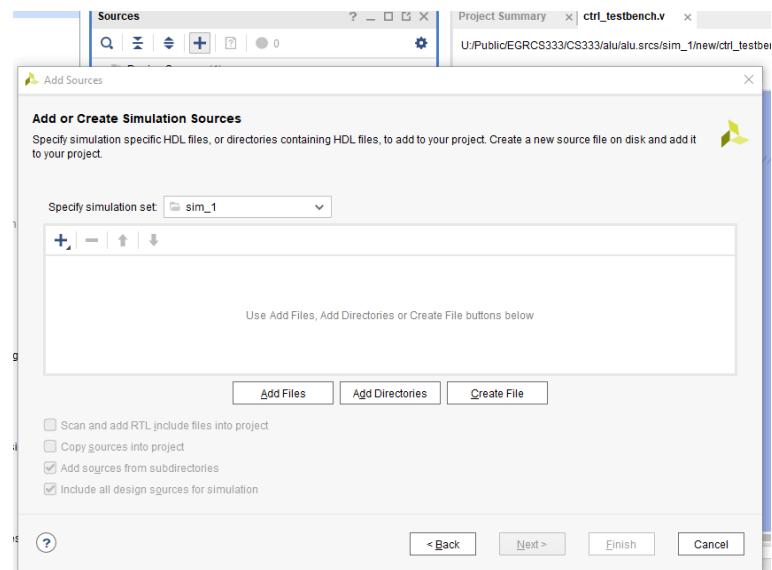
= is a blocking statement. In an always block, the line of code will be executed after the previous line has executed.

<= is non-blocking in nature. That means that in an always block, every line will be executed in parallel.

From Stack Overflow: [<https://stackoverflow.com/questions/35435420/what-is-the-difference-between-and-in-verilog>]

# 7 Test Benches & Timing Diagrams

Simulations show the output of a design without having to implement it to the board. These programs are called test benches and describe a set of steps that the simulator will execute on the design.



First, you must create a simulation source. Click the + icon in the Sources menu and select Simulation Source. Then create a new file.

```
reg[5:0] A;
reg[5:0] B;
reg[3:0] C;
wire[6:0] led;
reg clk;
```

Create registers for all the inputs and wires for all the outputs

```
initial begin
    clk = 0;
    forever #5 clk=~clk;
end
```

Create a new initial block, this sets the clk to 0 at first and every 5 nanoseconds the clock will flip. This is then set for the entire testbench

```
ctrl dut (
    .clk(clk),
    .A(A),
    .B(B),
    .C(C),
    .led(led)
);
```

After that set up your circuit. Here ctrl is the circuit being tested

```
initial begin
    A = 6'b000001;
    B = 6'b000001;
    C = 4'b0000;
    #100;
    A = 6'b000000;
    B = 6'b000000;
    C = 4'b0000;
    #100;
    A = 6'b000010;
    B = 6'b000010;
    C = 4'b0001;
end
```

Then you describe your test bench. Here the inputs are set to different values and then are set to wait 100 nanoseconds between changing again. This allows us to see the changes happening in the test bench

Now that the test bench is created, it's time to run the simulation, under the Simulation tab of the Project manager, click on Run Simulation and select Run Behavioral Simulation. After its finished click the Zoom to

Fit button  to put the test bench in view and you'll have the test bench



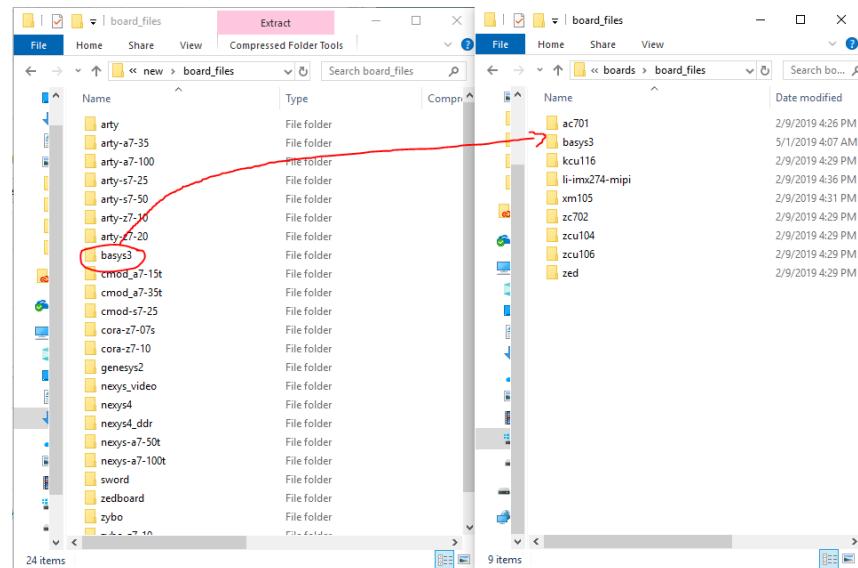
# 8 Using IP Cores

This section focuses on how to implement a basic design using IP cores in Vivado. IP Cores stand for **Intellectual Property** core. These allow for quickly adding already designed designs to a circuit. Note that this is very different from the previous chapters of the manual. We will no longer be programming in Verilog but creating a circuit by dragging and dropping components and allowing Vivado to do a lot of the work for us. Once we have designed our circuit, we'll write code in the Vivado SDK in which compilation is targeted to run on our designed device

This first part will walk you through how to create a **Microblaze** design in Vivado. Microblaze is a microprocessor core defined in software. It's a 32bit RISC based processor with many configurable variables. The Microblaze is a processor which we can design around and then write to the FPGA.

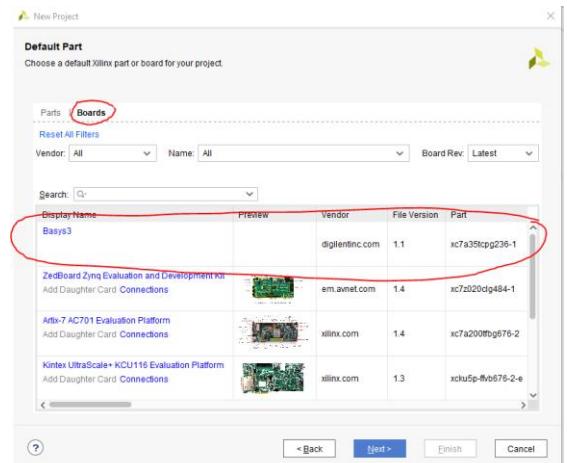
Before starting you need to obtain the Basys 3 board files from the Digilent Github repository [here](https://github.com/Digilent/vivado-boards) [<https://github.com/Digilent/vivado-boards>]. Download or clone the repository and browse to the board files of your Vivado install. Generally this is at C:\Xilinx\Vivado\20XX.X\data\boards\boardfiles. From the repository in /new/board\_files/ copy the Basys3 folder to your Vivado install directory.

This allows you to use the Basys 3 board in the Vivado IP Designer. If you had Vivado running during this process. You need to restart Vivado if it was running during that process



At this point create a Vivado RTL project like you usually would. Be sure not to define any sources.

At the stage where you would traditionally choose the part being used. Select the **Boards** tab and select the Basys 3



Complete the project creation as usual

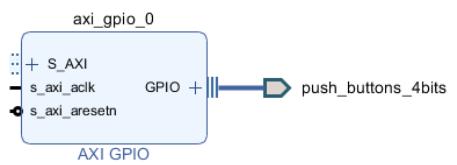
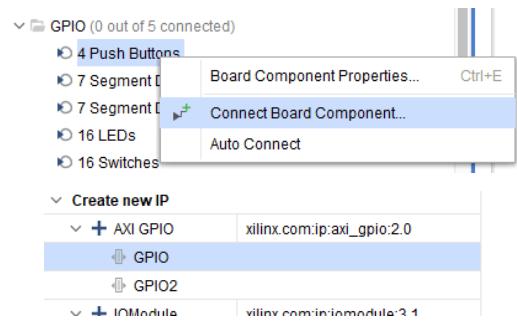
This is where the design process deviates from the traditional Verilog design. Instead of creating sources, open the IP Integrator tab and select **Create Block Design**. Give the design a name and click OK.

Be sure in the top left window the following tabs are present: **Sources**, **Design**, **Signals**, **Board**. If Board is missing, go back, you currently have the part still selected and need to choose the Basys 3 board. Now to begin the design

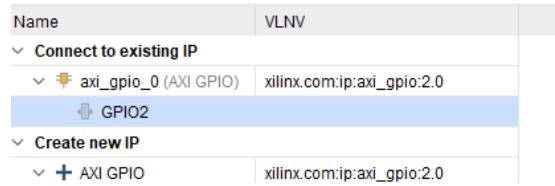
Open the board tab and find **GPIO** in the sections. Right click on **4 Push Buttons** and click **Connect Board Component...** This inserts the push buttons on the board into our IP Design.

In the dialog window that pops up select **GPIO** under the **AXI GPIO** section. Click OK

You should now see the GPIO block in the Diagram window with the push buttons connected



Now connect the **16 LEDs** and select **GPIO2** under the **Connect to existing IP** tab. Click OK

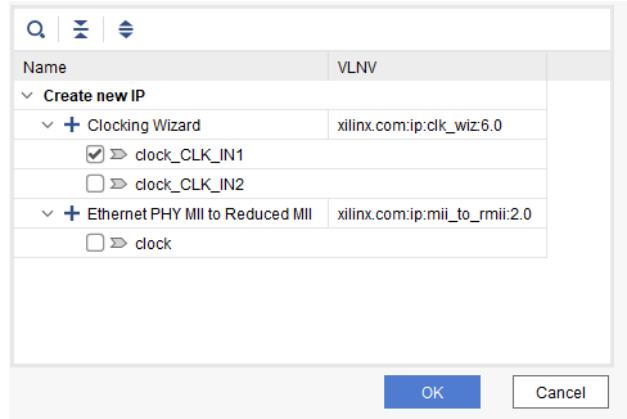


The GPIO block should now have a GPIO2 port and the LEDs at the end of it

Now we need to add the clock and the microprocessor itself.

In the boards tab add the **System Clock**.

In the first window make sure only **clock\_CLK\_IN1** is ticked



A new **Clocking Wizard** block will appear in your design. Double click on the block to open the customization window for the clock

In the **Board** tab, make sure the **CLK\_IN1** is set to **sys clock** and **EXT\_RESET\_IN** is set to **reset**

IP Interface	Board Interface
CLK_IN1	sys clock
CLK_IN2	Custom
EXT_RESET_IN	reset

In the **Output Clocks** tab make sure that **clk\_out1** both **checked** and set at **100Mhz** and at the bottom of the tab **Reset Type** is set to **Active High**

Output Clock	Port Name	Output Freq (MHz)		Phase (degrees)		Duty Cycle (%)		Drives	Use Fine PS	Max Freq. of buffer
		Requested	Actual	Requested	Actual	Requested	Actual			
<input checked="" type="checkbox"/> clk_out1	clk_out1	100.000	100.000	0.000	0.000	50.000	50.0	BUFG	▼	<input type="checkbox"/> 464.037
<input type="checkbox"/> clk_out2	clk_out2	100.000	N/A	0.000	N/A	50.000	N/A	BUFG	▼	<input type="checkbox"/> 464.037
<input type="checkbox"/> clk_out3	clk_out3	100.000	N/A	0.000	N/A	50.000	N/A	BUFG	▼	<input type="checkbox"/> 464.037

#### Reset Type

Active High  Active Low

At the top of the Diagram window there is a green bar click **Run Connection Automation** this does a bit of automatic configuring.

Now add the USB UART block from the Board window. This will allow for debugging over USB.

Now add the Microblaze microprocessor. To do this, right click on an empty spot of the diagram and click on **Add IP** in the window that pops up search for **Microblaze**. Another green ribbon will appear, click on **Run Block Automation** this will put all the blocks in the right place.

During this configuration make sure the follow values are as follows:

**Preset:** None

**Local Memory:** 32KB

**Local Memory ECC:** None

**Cache Configuration:** None

**Debug Module:** Debug Only

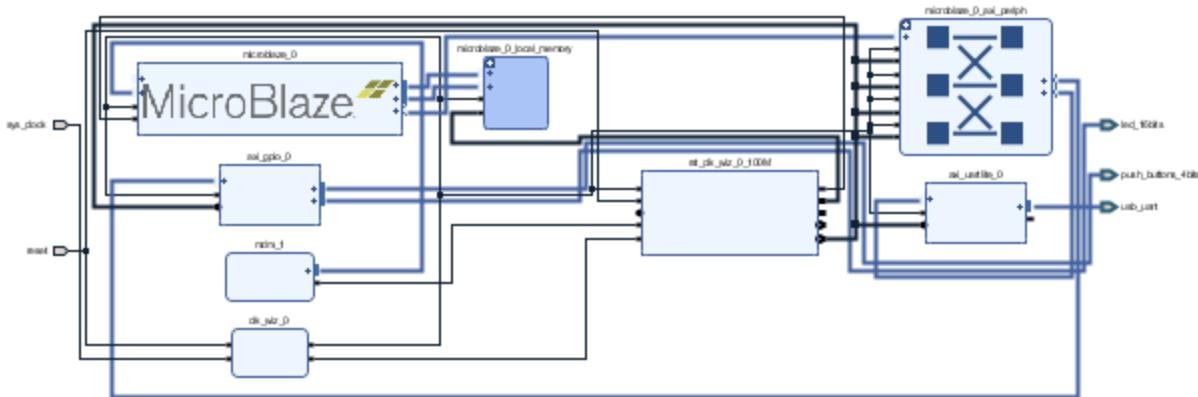
**Peripheral AXI Port:** Enabled

**Interrupt Controller:** Unchecked

**Clock Connection:** /clk\_wiz\_0/clk\_out1

Another green ribbon will appear, click **Run Connection Automation** then click OK

Hit **F6** to validate the design. If Validation is not successful, go back and restart the steps with a new design.



The final design should look something like this

The final thing to do before generating a bitstream is to generate an HDL wrapper to run the design on the board.

Switch to the **Sources** tab, right click on your design and select **Generate HDL Wrapper**. Tick **Let Vivado manage wrapper and auto-update** then click OK. This will generate your top file. Now click **Generate Bitstream**.

Now we need to export the design to the Vivado SDK. In the **File** menu select **Export** and then **Export Hardware**. Make sure **Include Bitstream** is ticked and keep the export location as the local project. Click OK

In the File Menu click **Launch SDK** make sure the locations are set to local project

The Vivado SDK is a software development environment in which you can develop code that is targeted at your design made in the IP integrator. The SDK can take a few minutes to load, please be patient. Once the SDK is loaded it will probably look very familiar, this is because the SDK is based in Eclipse. Everything language independent that you can do in eclipse you can do here.

Create a new Application Project in **File -> New -> Application Project**

Give your Application a name and make sure **C** is selected as the target language. Click next and choose **Empty Application** as the template. Click Finish.

Right click on the **src** folder and create a new source file. Call this file **main.c** add the

```
main.c

#include "xgpio.h"

//send data over UART
#include "xil_printf.h"

//information about AXI peripherals
#include "xparameters.h"

int main()
{
    XGpio gpio;
    u32 btn, led;

    XGpio_Initialize(&gpio, 0);

    XGpio_SetDataDirection(&gpio, 2, 0x00000000);
    XGpio_SetDataDirection(&gpio, 1, 0xFFFFFFFF);

    while (1)
    {
        btn = XGpio_DiscreteRead(&gpio, 1);

        if (btn != 0) // turn all LEDs on when any button is pressed
            led = 0xFFFFFFFF;
        else
            led = 0x00000000;

        XGpio_DiscreteWrite(&gpio, 2, led);

        xil_printf("\rbutton state: %08x", btn);
    }
}
```

following code to main.c

Now click on **Xilinx** in the toolbar and select **Program FPGA**

Keep all the settings the same and click **Program**

This will then load the program onto the FPGA, but not run it. Right click on your project folder and select **Run As -> Launch on Hardware (System Debugger)**

You will now need to obtain a serial console application. [Terra Term](https://osdn.net/projects/ttssh2/downloads/70691/teraterm-4.102.exe/) [<https://osdn.net/projects/ttssh2/downloads/70691/teraterm-4.102.exe/>] is recommended

Set TerraTerm to **serial** and select the COM port the FPGA is connected to. In the serial port setup make sure to set the following settings

**Speed:** 9600

**Data:** 8-Bit

**Parity:** None

**Stop Bits:** 1 bit

You should now be able to see the message in the terminal and see the hex representation of the buttons in the terminal

COM4 - Tera Term VT

File Edit Setup Control Window Help

```
button state: 00000000
```

COM4 - Tera Term VT

File Edit Setup Control Window Help

```
button state: 00000002
```

# 9 Using Pmod (Compass Example)

PMod is an open standard developed by Digilent to use with FPGAs or Microcontrollers. As of the writing of this manual we have the **CMPS2** a compass module, the **DHB1** a motor controller, and **GPS** which does exactly what you think it does. In the coming sections we will cover how to use all of these modules. Each of these also uses their own connection method, so parts of these may be able to be used with other Pmod devices.

To use these, insert them into the PMod ports on the sides of the FPGA

Before programming anything we need to get the Digilent IP Library from their repository [here](#)

[[https://github.com/Digilent/vivado-library/releases?\\_ga=2.10682993.418939316.1557554693-1032257056.1556697100](https://github.com/Digilent/vivado-library/releases?_ga=2.10682993.418939316.1557554693-1032257056.1556697100)]

Download the latest release even if you are using a new version of Vivado than what is shown. For example while writing this manual I am using 2018.3 while the newest version of their library is 2018.2. Extract it somewhere where you can find it later

Open the project you made in the last section.

We need to add the Digilent Library you just downloaded, under **Project Manager** open **Settings**

Click the arrow next to **IP** and select **Repository**

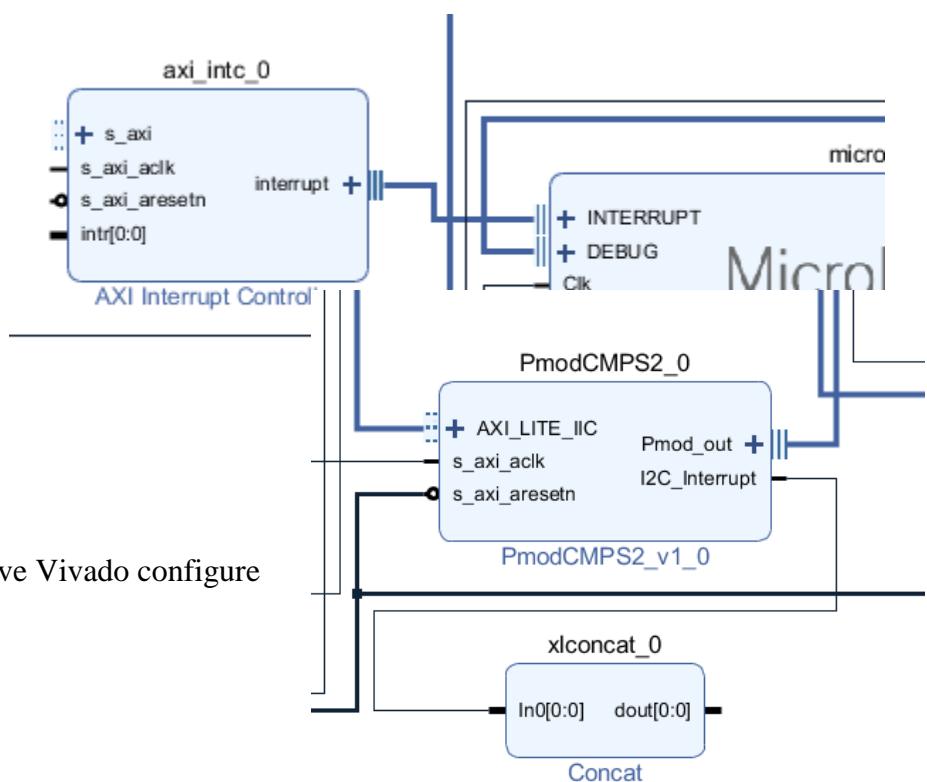
Click the + and add the extracted repository. A window should come up telling you that IP cores have been added. Click **Apply** and **OK**

In the **Board** tab scroll down to the Pmod Connectors. Connect it to your design and select the Pmod component you want to use. In this first part we'll be using the compass module **CMPS2**. In this example I have CMPS2 connected to Pmod port JA.

Run **Connection Automation** make sure your Pmod device is checked.

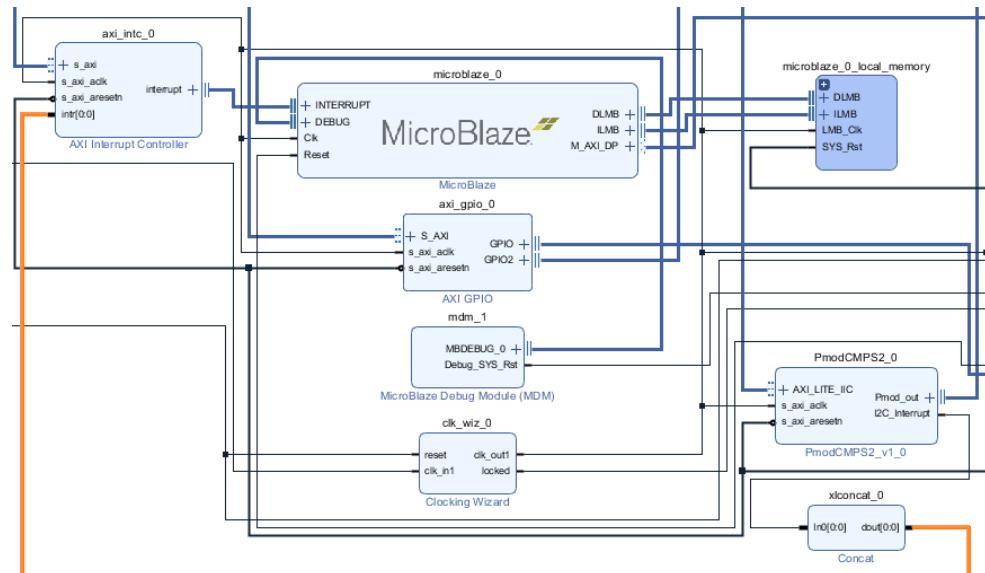
At this point 2 blocks may need to be added another **clock** or a Microblaze **interrupt**. The CMPS2 does not need a clock, but it does need the interrupt. The clock will not be covered in this manual since none of our components require a separate clock.

We now need to add an **AXI Interrupt Controller**. Add the AXI Interrupt Controller to your design and manually connect the interrupt port on that block to the one of the Microblaze processor.



Now click **Run Connection Automation** to have Vivado configure everything else.

Now add a **Concat** IP Core. Reconfigure it to only have the number of Pmod devices you are using (right now we are only using 1). Now connect the Interrupt port on your Pmod Device to the In port of the concat block.



Now connect the concat output port to the **intr** port of the AXI Interrupt Controller.

Click the Regenerate Layout Button  this organizes the layout of the design a bit

And then validate the design. If validation is not successful restart these steps, you missed a step or didn't configure a component correctly.

Return to the sources window and re-generate the HDL wrapper and generate a bitstream. Export and open the SDK like you did in the previous section.

Create a new Empty Project, make sure C is the target language

Under the platform folder at the top, expand the folder and the driver folder. In here you can find the Pmod interface and an example

Move the example source file you want into the src folder of your project.

Write the program to the FPGA and run as you did in the previous step.

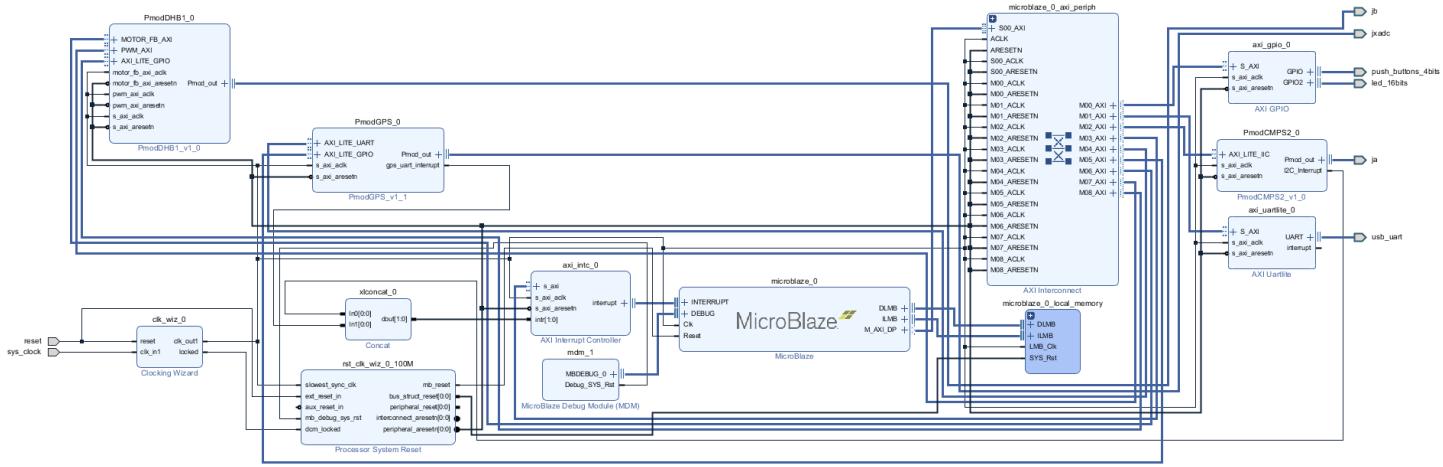
To show information over the Serial Terminal, change any instances of `printf` to `xil_printf` this tells the FPGA to write to the serial terminal.

If all works you should receive compass info (or info from your FPGA perfectly)

You can either continue working off the same design as in previous sections or create a new one. I will be building off the ones in previous design so that the finished design will have motor control, GPS and compass in it.

To implement the other devices to your design, add new Pmod blocks to the design. Be sure to increase the concat if it needs an Interrupt and connect that and make a new clock if your Pmod requires it. Run the connection wizard after you are done

The GPS, Motor, and Compass should look something like this



Be sure to generate the HDL Wrapper and then export the hardware to the SDK. Using the examples given you can then implement the other 2 Pmod devices.



# Appendix

This appendix holds information useful to the manual that doesn't belong in the middle of a section

## Example Verilog Header Comment

```
//////////  
// Company: Elizabethtown College  
// Engineer: Clay Buxton  
//  
// Create Date: 02/09/2019 08:02:42 PM  
// Design Name: 4 Bit Adder  
// Module Name: top  
// Project Name: 4 Bit Adder  
// Target Devices: Basys 3  
// Tool Versions: Vivado 2018.3  
// Description: Adds 4-bit numbers  
//  
// Dependencies: fulladder.v  
//  
// Revision: 0.02 - Project Finished  
// Revision 0.01 - File Created  
// Additional Comments: None  
//  
//////////
```



## Description

The Arduino Uno R3 is the perfect board to get familiar with electronics and coding. This versatile microcontroller is equipped with the well-known ATmega328P and the ATMega 16U2 Processor.

This board will give you a great first experience within the world of Arduino.

## Target areas:

Maker, introduction, industries



## Features

- **ATMega328P Processor**

- **Memory**

- AVR CPU at up to 16 MHz
    - 32KB Flash
    - 2KB SRAM
    - 1KB EEPROM

- **Security**

- Power On Reset (POR)
    - Brown Out Detection (BOD)

- **Peripherals**

- 2x 8-bit Timer/Counter with a dedicated period register and compare channels
    - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
    - 1x USART with fractional baud rate generator and start-of-frame detection
    - 1x controller/peripheral Serial Peripheral Interface (SPI)
    - 1x Dual mode controller/peripheral I2C
    - 1x Analog Comparator (AC) with a scalable reference input
    - Watchdog Timer with separate on-chip oscillator
    - Six PWM channels
    - Interrupt and wake-up on pin change

- **ATMega16U2 Processor**

- 8-bit AVR® RISC-based microcontroller

- **Memory**

- 16 KB ISP Flash
    - 512B EEPROM
    - 512B SRAM
    - debugWIRE interface for on-chip debugging and programming

- **Power**

- 2.7-5.5 volts



# CONTENTS

<b>1 The Board</b>	<b>4</b>
1.1 Application Examples	4
1.2 Related Products	4
<b>2 Ratings</b>	<b>4</b>
2.1 Recommended Operating Conditions	4
2.2 Power Consumption	5
<b>3 Functional Overview</b>	<b>5</b>
3.1 Board Topology	5
3.2 Processor	6
3.3 Power Tree	6
<b>4 Board Operation</b>	<b>7</b>
4.1 Getting Started - IDE	7
4.2 Getting Started - Arduino Web Editor	7
4.3 Getting Started - Arduino IoT Cloud	7
4.4 Sample Sketches	7
4.5 Online Resources	7
<b>5 Connector Pinouts</b>	<b>8</b>
5.1 JANALOG	9
5.2 JDIGITAL	9
5.3 Mechanical Information	10
5.4 Board Outline & Mounting Holes	10
<b>6 Certifications</b>	<b>11</b>
6.1 Declaration of Conformity CE DoC (EU)	11
6.2 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021	11
6.3 Conflict Minerals Declaration	12
<b>7 FCC Caution</b>	<b>12</b>
<b>8 Company Information</b>	<b>13</b>
<b>9 Reference Documentation</b>	<b>13</b>
<b>10 Revision History</b>	<b>13</b>



## 1 The Board

### 1.1 Application Examples

The UNO board is the flagship product of Arduino. Regardless if you are new to the world of electronics or will use the UNO as a tool for education purposes or industry-related tasks.

**First entry to electronics:** If this is your first project within coding and electronics, get started with our most used and documented board; Arduino UNO. It is equipped with the well-known ATmega328P processor, 14 digital input/output pins, 6 analog inputs, USB connections, ICSP header and reset button. This board includes everything you will need for a great first experience with Arduino.

**Industry-standard development board:** Using the Arduino UNO board in industries, there are a range of companies using the UNO board as the brain for their PLC's.

**Education purposes:** Although the UNO board has been with us for about ten years, it is still widely used for various education purposes and scientific projects. The board's high standard and top quality performance makes it a great resource to capture real time from sensors and to trigger complex laboratory equipment to mention a few examples.

### 1.2 Related Products

- Starter Kit
- Tinkerkit Braccio Robot
- Example

## 2 Ratings

### 2.1 Recommended Operating Conditions

Symbol	Description	Min	Max
	Conservative thermal limits for the whole board:	-40 °C (-40°F)	85 °C ( 185°F)

**NOTE:** In extreme temperatures, EEPROM, voltage regulator, and the crystal oscillator, might not work as expected due to the extreme temperature conditions

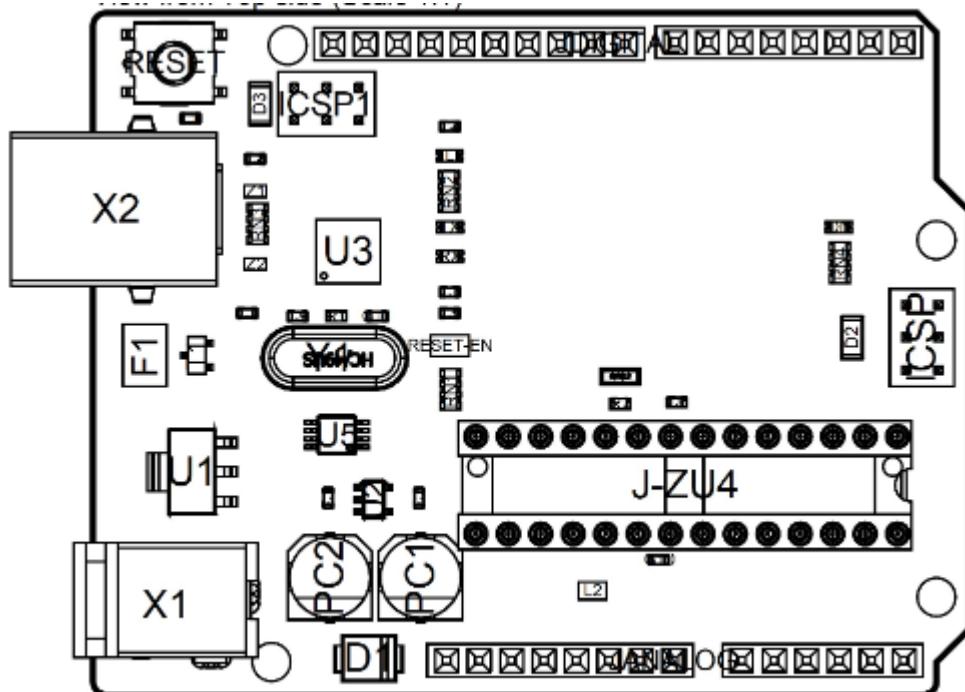
## 2.2 Power Consumption

Symbol	Description	Min	Typ	Max	Unit
VINMax	Maximum input voltage from VIN pad	6	-	20	V
VUSBMax	Maximum input voltage from USB connector		-	5.5	V
PMax	Maximum Power Consumption	-	-	xx	mA

## 3 Functional Overview

### 3.1 Board Topology

Top view



Board topology

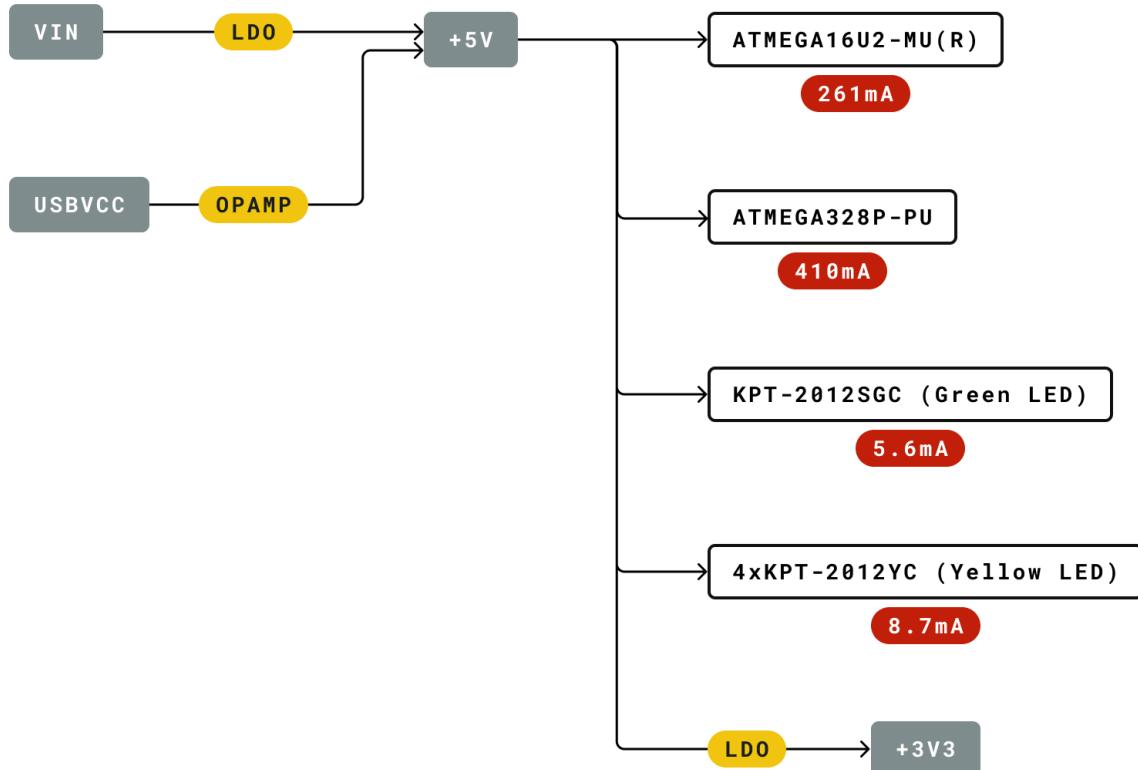
Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		



### 3.2 Processor

The Main Processor is a ATmega328P running at up tp 20 MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the USB Bridge coprocessor.

### 3.3 Power Tree



**Legend:**

- |                                    |  |   |
|------------------------------------|--|---|
| <input type="checkbox"/> Component | <span style="color: #ccc;">●</span> Power I/O  | <span style="color: yellow;">●</span> Conversion Type |
|                                    |  |   |
|                                    | <span style="color: red;">●</span> Max Current | <span style="color: teal;">●</span> Voltage Range     |

*Power tree*



## 4 Board Operation

### 4.1 Getting Started - IDE

If you want to program your Arduino UNO while offline you need to install the Arduino Desktop IDE [1] To connect the Arduino UNO to your computer, you'll need a Micro-B USB cable. This also provides power to the board, as indicated by the LED.

### 4.2 Getting Started - Arduino Web Editor

All Arduino boards, including this one, work out-of-the-box on the Arduino Web Editor [2], by just installing a simple plugin.

The Arduino Web Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow [3] to start coding on the browser and upload your sketches onto your board.

### 4.3 Getting Started - Arduino IoT Cloud

All Arduino IoT enabled products are supported on Arduino IoT Cloud which allows you to Log, graph and analyze sensor data, trigger events, and automate your home or business.

### 4.4 Sample Sketches

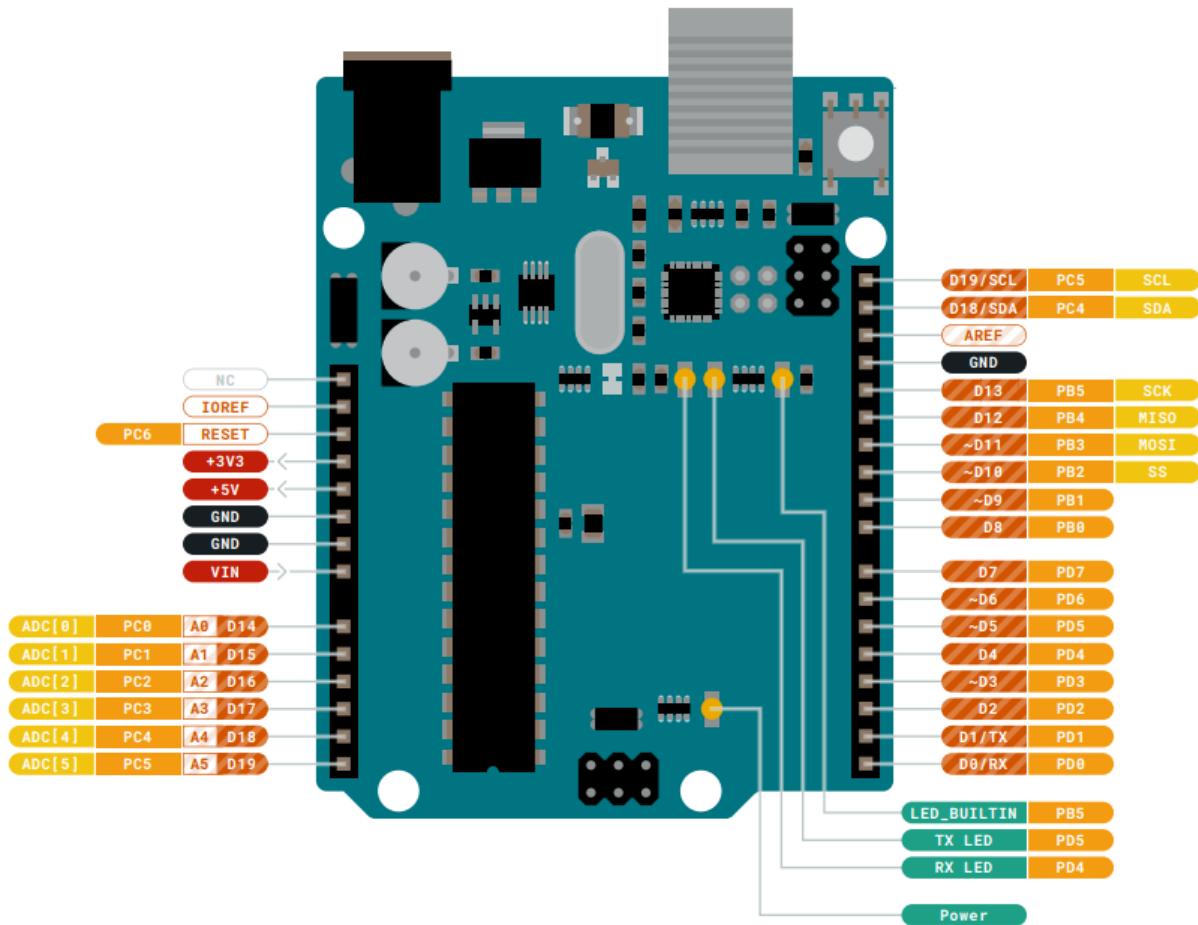
Sample sketches for the Arduino XXX can be found either in the "Examples" menu in the Arduino IDE or in the "Documentation" section of the Arduino Pro website [4]

### 4.5 Online Resources

Now that you have gone through the basics of what you can do with the board you can explore the endless possibilities it provides by checking exciting projects on ProjectHub [5], the Arduino Library Reference [6] and the online store [7] where you will be able to complement your board with sensors, actuators and more



## 5 Connector Pinouts



Pinout



## 5.1 JANALOG

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

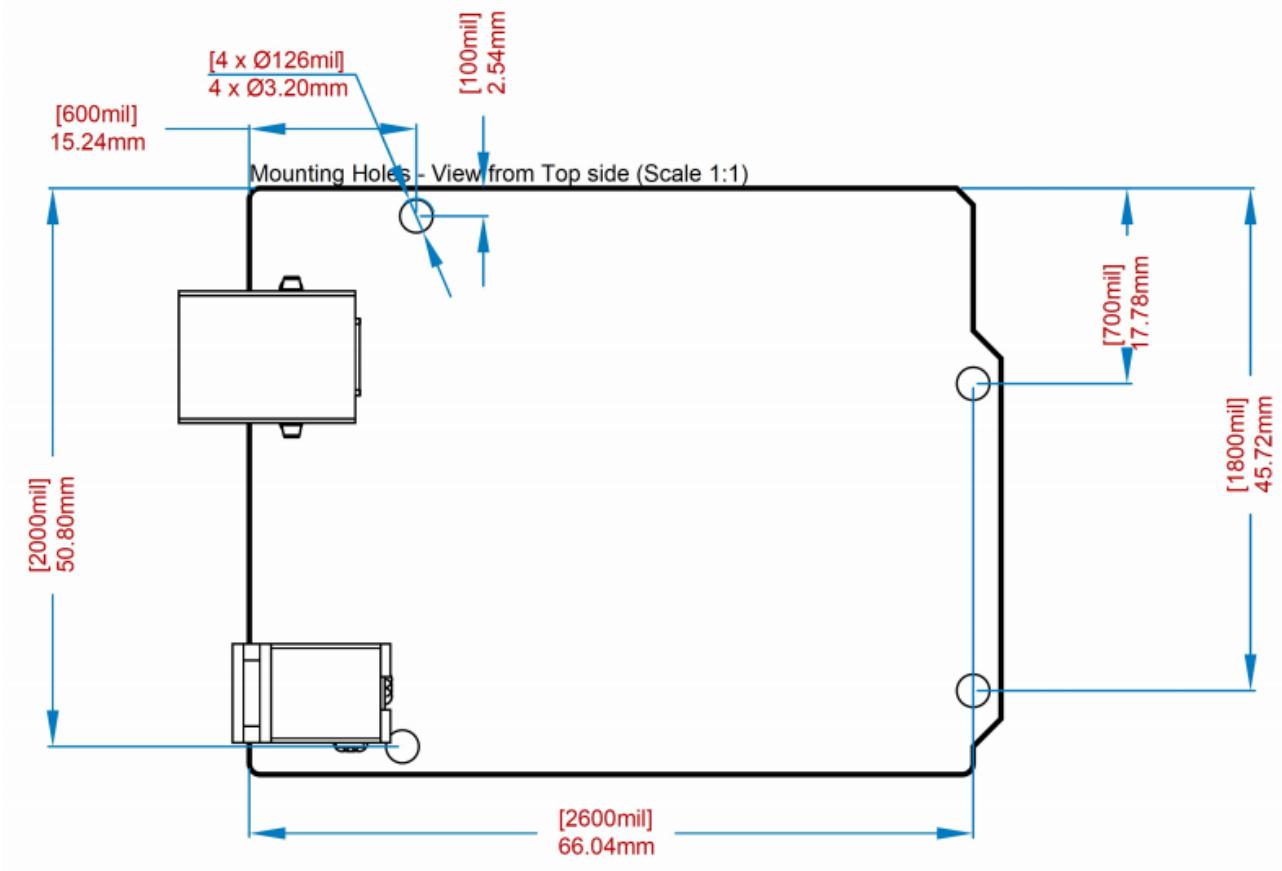
## 5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)



## 5.3 Mechanical Information

## 5.4 Board Outline & Mounting Holes





## 6 Certifications

### 6.1 Declaration of Conformity CE DoC (EU)

We declare under our sole responsibility that the products above are in conformity with the essential requirements of the following EU Directives and therefore qualify for free movement within markets comprising the European Union (EU) and European Economic Area (EEA).

<b>ROHS 2 Directive 2011/65/EU</b>	
Conforms to:	EN50581:2012
<b>Directive 2014/35/EU. (LVD)</b>	
Conforms to:	EN 60950-1:2006/A11:2009/A1:2010/A12:2011/AC:2011
<b>Directive 2004/40/EC &amp; 2008/46/EC &amp; 2013/35/EU, EMF</b>	
Conforms to:	EN 62311:2008

### 6.2 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021

Arduino boards are in compliance with RoHS 2 Directive 2011/65/EU of the European Parliament and RoHS 3 Directive 2015/863/EU of the Council of 4 June 2015 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

<b>Substance</b>	<b>Maximum limit (ppm)</b>
Lead (Pb)	1000
Cadmium (Cd)	100
Mercury (Hg)	1000
Hexavalent Chromium (Cr6+)	1000
Poly Brominated Biphenyls (PBB)	1000
Poly Brominated Diphenyl ethers (PBDE)	1000
Bis(2-Ethylhexyl) phthalate (DEHP)	1000
Benzyl butyl phthalate (BBP)	1000
Dibutyl phthalate (DBP)	1000
Diisobutyl phthalate (DIBP)	1000

Exemptions: No exemptions are claimed.

Arduino Boards are fully compliant with the related requirements of European Union Regulation (EC) 1907 /2006 concerning the Registration, Evaluation, Authorization and Restriction of Chemicals (REACH). We declare none of the SVHCs (<https://echa.europa.eu/web/guest/candidate-list-table>), the Candidate List of Substances of Very High Concern for authorization currently released by ECHA, is present in all products (and also package) in quantities totaling in a concentration equal or above 0.1%. To the best of our knowledge, we also declare that our products do not contain any of the substances listed on the "Authorization List" (Annex XIV of the REACH regulations) and Substances of Very High Concern (SVHC) in any significant amounts as specified by the Annex XVII of Candidate list published by ECHA (European Chemical Agency) 1907 /2006/EC.



### 6.3 Conflict Minerals Declaration

As a global supplier of electronic and electrical components, Arduino is aware of our obligations with regards to laws and regulations regarding Conflict Minerals, specifically the Dodd-Frank Wall Street Reform and Consumer Protection Act, Section 1502. Arduino does not directly source or process conflict minerals such as Tin, Tantalum, Tungsten, or Gold. Conflict minerals are contained in our products in the form of solder, or as a component in metal alloys. As part of our reasonable due diligence Arduino has contacted component suppliers within our supply chain to verify their continued compliance with the regulations. Based on the information received thus far we declare that our products contain Conflict Minerals sourced from conflict-free areas.

## 7 FCC Caution

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

- (1) This device may not cause harmful interference
- (2) this device must accept any interference received, including interference that may cause undesired operation.

**FCC RF Radiation Exposure Statement:**

1. This Transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
2. This equipment complies with RF radiation exposure limits set forth for an uncontrolled environment.
3. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

English: User manuals for license-exempt radio apparatus shall contain the following or equivalent notice in a conspicuous location in the user manual or alternatively on the device or both. This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions:

- (1) this device may not cause interference
- (2) this device must accept any interference, including interference that may cause undesired operation of the device.

French: Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

- (1) l'appareil ne doit pas produire de brouillage
- (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

**IC SAR Warning:**

English This equipment should be installed and operated with minimum distance 20 cm between the radiator and your body.

French: Lors de l' installation et de l' exploitation de ce dispositif, la distance entre le radiateur et le corps est d 'au moins 20 cm.



**Important:** The operating temperature of the EUT can't exceed 85°C and shouldn't be lower than -40°C.

Hereby, Arduino S.r.l. declares that this product is in compliance with essential requirements and other relevant provisions of Directive 2014/53/EU. This product is allowed to be used in all EU member states.

## 8 Company Information

<b>Company name</b>	Arduino S.r.l
Company Address	Via Andrea Appiani 25 20900 MONZA Italy

## 9 Reference Documentation

Reference	Link
Arduino IDE (Desktop)	<a href="https://www.arduino.cc/en/Main/Software">https://www.arduino.cc/en/Main/Software</a>
Arduino IDE (Cloud)	<a href="https://create.arduino.cc/editor">https://create.arduino.cc/editor</a>
Cloud IDE Getting Started	<a href="https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-4b3e4a">https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-4b3e4a</a>
Arduino Pro Website	<a href="https://www.arduino.cc/pro">https://www.arduino.cc/pro</a>
Project Hub	<a href="https://create.arduino.cc/projecthub?by=part&amp;part_id=11332&amp;sort=trending">https://create.arduino.cc/projecthub?by=part&amp;part_id=11332&amp;sort=trending</a>
Library Reference	<a href="https://www.arduino.cc/reference/en/">https://www.arduino.cc/reference/en/</a>
Online Store	<a href="https://store.arduino.cc/">https://store.arduino.cc/</a>

## 10 Revision History

Date	Revision	Changes
xx/06/2021	1	Datasheet release

## Basys 3™ FPGA Board Reference Manual

Revised April 8, 2016

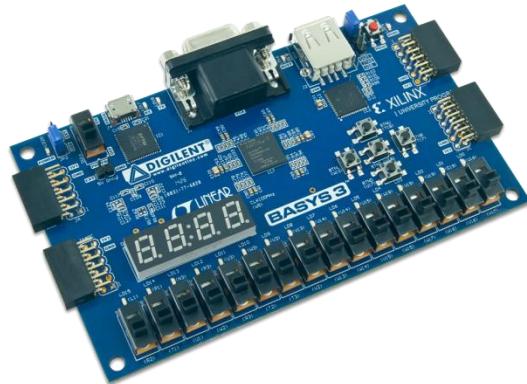
This manual applies to the Basys 3 rev. C

---

## Overview

The Basys 3 board is a complete, ready-to-use digital circuit development platform based on the latest Artix®-7 Field Programmable Gate Array (FPGA) from Xilinx®. With its high-capacity FPGA (Xilinx part number XC7A35T-1CPG236C), low overall cost, and collection of USB, VGA, and other ports, the Basys 3 can host designs ranging from introductory combinational circuits to complex sequential circuits like embedded processors and controllers. It includes enough switches, LEDs, and other I/O devices to allow a large number of designs to be completed without the need for any additional hardware, and enough uncommitted FPGA I/O pins to allow designs to be expanded using Digilent Pmods or other custom boards and circuits.

The Artix-7 FPGA is optimized for high performance logic, and offers more capacity, higher performance, and more resources than earlier designs. Artix-7 35T features include:



- 33,280 logic cells in 5200 slices (each slice contains four 6-input LUTs and 8 flip-flops)
- 1,800 Kbits of fast block RAM
- Five clock management tiles, each with a phase-locked loop (PLL)
- 90 DSP slices
- Internal clock speeds exceeding 450MHz
- On-chip analog-to-digital converter (XADC)

*The Basys 3.*

The Basys 3 also offers an improved collection of ports and peripherals, including:

- |   |  |                         |
|---|--|-------------------------|
| • 16 user switches  | • 16 user LEDs                                       | • 5 user pushbuttons    |
| • 4-digit 7-segment display                                     | • Three Pmod ports                                   | • Pmod for XADC signals |
| • 12-bit VGA output   | • USB-UART Bridge                                    | • Serial Flash          |
| • Digilent USB-JTAG port for FPGA programming and communication | • USB HID Host for mice, keyboards and memory sticks |                         |

The Basys 3 works with Xilinx's new high-performance Vivado™ Design Suite. Vivado includes many new tools and design flows that facilitate and enhance the latest design methods. It runs faster, allows better use of FPGA resources, and allows designers to focus their time evaluating design alternatives. The System Edition includes an on-chip logic analyzer, high-level synthesis tool, other cutting-edge tools, and the free WebPACK™ version allows Basys 3 designs to be created at no additional cost.

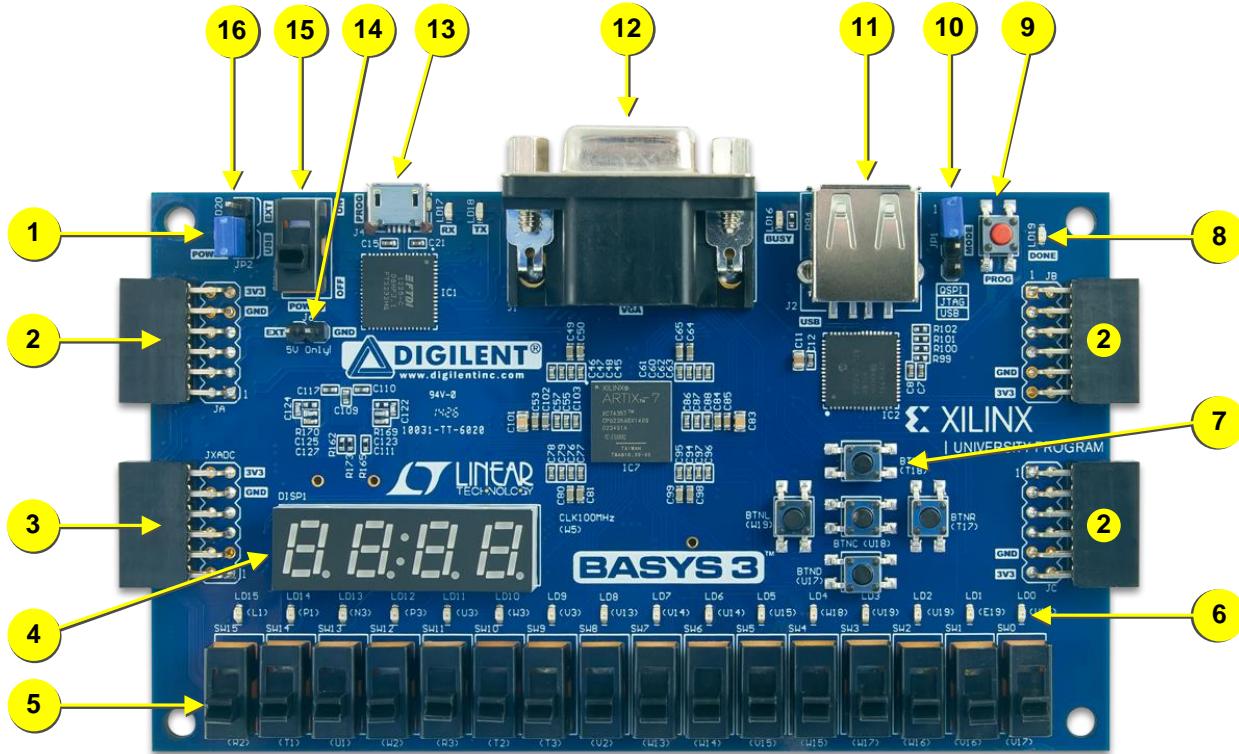


Figure 1. Basys 3 FPGA board with callouts.

Callout	Component Description	Callout	Component Description
1	Power good LED	9	FPGA configuration reset button
2	Pmod port(s)	10	Programming mode jumper
3	Analog signal Pmod port (XADC)	11	USB host connector
4	Four digit 7-segment display	12	VGA connector
5	Slide switches (16)	13	Shared UART/ JTAG USB port
6	LEDs (16)	14	External power connector
7	Pushbuttons (5)	15	Power Switch
8	FPGA programming done LED	16	Power Select Jumper

Table 1. Basys 3 Callouts and component descriptions.

A growing collection of board support IP, reference designs, and add-on boards are available on the Digilent website. See the Basys 3 page at [www.digilentinc.com](http://www.digilentinc.com) for more information.

## 1 Power Supplies

The Basys 3 board can receive power from the Digilent USB-JTAG port (J4) or from a 5V external power supply. Jumper JP3 (near the power switch) determines which source is used.

All Basys 3 power supplies can be turned on and off by a single logic-level power switch (SW16). A power-good LED (LD20), driven by the "power good" output of the LTC3633 supply, indicates that the supplies are turned on and operating normally. An overview of the Basys 3 power circuit is shown in Fig. 2.

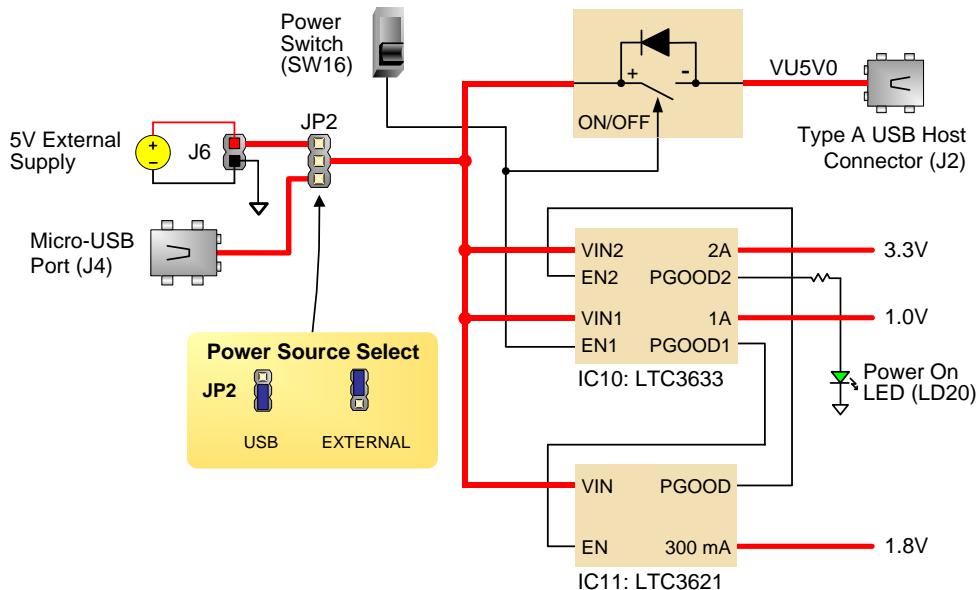


Figure 2. Basys 3 power circuit.

The USB port can deliver enough power for the vast majority of designs. A few demanding applications, including any that drive multiple peripheral boards, might require more power than the USB port can provide. Also, some applications may need to run without being connected to a PC's USB port. In these instances an external power supply or battery pack can be used.

An external power supply can be used by plugging into the external power header (J6) and setting jumper JP2 to "EXT". The supply must deliver 4.5VDC to 5.5VDC and at least 1A of current (i.e., at least 5W of power). Many suitable supplies can be purchased through Digi-Key or other catalog vendors.

An external battery pack can be used by connecting the battery's positive terminal to the "EXT" pin of J6 and the negative terminal to the "GND" pin of J6. The power provided to USB devices that are connected to Host connector J2 is not regulated. Therefore, it is necessary to limit the maximum voltage of an external battery pack to 5.5V DC. The minimum voltage of the battery pack depends on the application; if the USB Host function (J2) is used, at least 4.6V needs to be provided. In other cases, the minimum voltage is 3.6V.

Voltage regulator circuits from Linear Technology create the required 3.3V, 1.8V, and 1.0V supplies from the main power input. Table 2 provides additional information (typical currents depend strongly on FPGA configuration and the values provided are typical of medium size/speed designs).

Supply	Circuits	Device	Current (max/typical)
3.3V	FPGA I/O, USB ports, Clocks, Flash, PMODs	IC10: LTC3633	2A/0.1 to 1.5A
1.0V	FPGA Core	IC10: LTC3633	2A/ 0.2 to 1.3A
1.8V	FPGA Auxiliary and Ram	IC11: LTC3621	300mA/ 0.05 to 0.15A

Table 2. Basys 3 power supplies.

## 2 FPGA Configuration

After power-on, the Artix-7 FPGA must be configured (or programmed) before it can perform any functions. You can configure the FPGA in one of three ways:

1. A PC can use the Digilent USB-JTAG circuitry (port J4, labeled "PROG") to program the FPGA any time the power is on.
2. A file stored in the nonvolatile serial (SPI) flash device can be transferred to the FPGA using the SPI port.
3. A programming file can be transferred from a USB memory stick attached to the USB HID port.

Figure 3 shows the different options available for configuring the FPGA. An on-board "mode" jumper (JP1) selects between the programming modes.

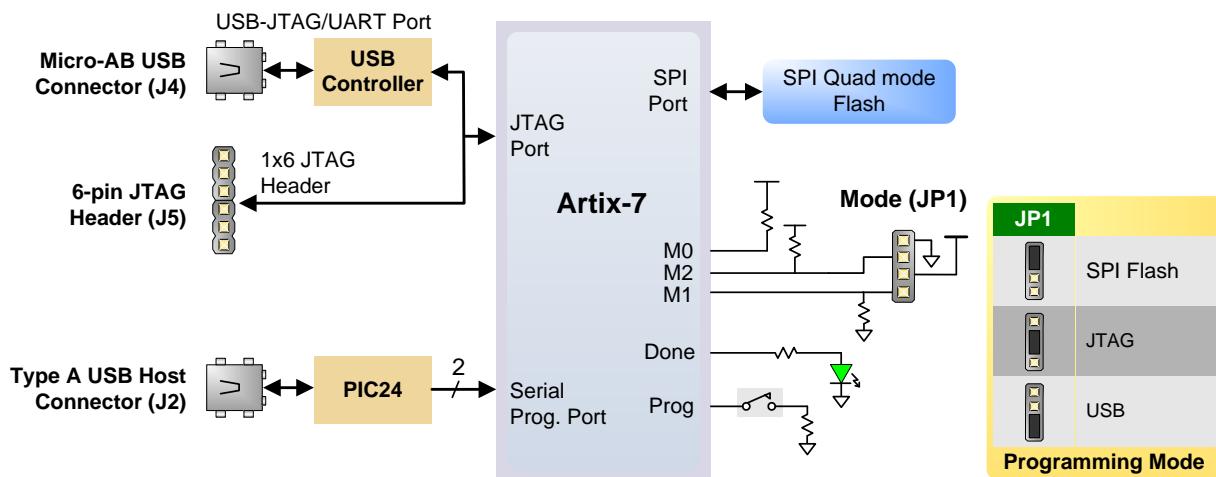


Figure 3. Basys 3 configuration options.

The FPGA configuration data is stored in files called bitstreams that have the .bit file extension. The Vivado software from Xilinx can create bitstreams from VHDL, Verilog®, or schematic-based source files.

Bitstreams are stored in SRAM-based memory cells within the FPGA. This data defines the FPGA's logic functions and circuit connections, and it remains valid until it is erased by removing board power, by pressing the reset button attached to the PROG input, or by writing a new configuration file using the JTAG port.

An Artix-7 35T bitstream is typically 17,536,096 bits and can take a long time to transfer. The time it takes to program the Basys 3 can be decreased by compressing the bitstream before programming, and then allowing the FPGA to decompress the bitstream itself during configuration. Depending on design complexity, compression ratios of 10x can be achieved. Bitstream compression can be enabled within the Xilinx Tools (Vivado) to occur during generation. For instructions on how to do this, consult the Xilinx documentation for the toolset being used.

After being successfully programmed, the FPGA will cause the "DONE" LED to illuminate. Pressing the "PROG" button at any time will reset the configuration memory in the FPGA. After being reset, the FPGA will immediately attempt to reprogram itself from whatever method has been selected by the programming mode jumper.

The following sections provide greater detail about programming the Basys 3 using the different methods available.

## 2.1 JTAG Programming

The Xilinx Tools typically communicate with FPGAs using the Test Access Port and Boundary-Scan Architecture, commonly referred to as JTAG. During JTAG programming, a .bit file is transferred from the PC to the FPGA using the onboard Digilent USB-JTAG circuitry (port J4) or an external JTAG programmer, such as the Digilent JTAG-HS2 attached to port J5 (located below port JA). You can perform JTAG programming any time after the Basys 3 has been powered on regardless of what the mode jumper (JP1) is set to. If the FPGA is already configured, then the existing configuration is overwritten with the bitstream being transmitted over JTAG. Setting the mode jumper to the JTAG setting (seen in Fig. 3) is useful to prevent the FPGA from being configured from any other bitstream source until a JTAG programming occurs.

Programming the Basys 3 with an uncompressed bitstream using the on-board USB\_JTAG circuitry usually takes around five seconds. JTAG programming can be done using the hardware server in Vivado. The demonstration project available at [digilentinc.com](http://digilentinc.com) provides an in-depth tutorial on how to program your board.

## 2.2 JTAG Programming

When programming a nonvolatile flash device, a bitstream file is transferred to the flash in a two-step process. First, the FPGA is programmed with a circuit that can program flash devices, and then data is transferred to the flash device via the FPGA circuit (this complexity is hidden from the user by the Xilinx Tools). After the flash device has been programmed, it can automatically configure the FPGA at a subsequent power-on or reset event as determined by the mode jumper setting (see Fig. 3). Programming files stored in the flash device will remain until they are overwritten, regardless of power-cycle events.

Programming the flash can take as long as one or two minutes, which is mostly due to the lengthy erase process inherent to the memory technology. Once written, however, FPGA configuration can be very fast – less than a second. Bitstream compression, SPI bus width, and configuration rate are factors controlled by the Xilinx Tools that can affect configuration speed.

Quad-SPI programming can be performed using Vivado.

## 2.3 USB Host Programming

You can program the FPGA from a pen drive attached to the USB-HID port (J2) by doing the following:

1. Format the storage device (Pen drive) with a FAT32 file system.
2. Place a single .bit configuration file in the root directory of the storage device.
3. Attach the storage device to the Basys 3.
4. Set the JP1 Programming Mode jumper on the Basys 3 to "USB".
5. Push the PROG button or power-cycle the Basys 3.

The FPGA will automatically be configured with the .bit file on the selected storage device. Any .bit files that are not built for the proper Artix-7 device will be rejected by the FPGA.

The Auxiliary Function Status, or "BUSY" LED (LD16), gives visual feedback on the state of the configuration process when the FPGA is not yet programmed:

- When steadily lit, the auxiliary microcontroller is either booting up or currently reading the configuration medium (pen drive) and downloading a bitstream to the FPGA.
- A slow pulse means the microcontroller is waiting for a configuration medium to be plugged in.
- In case of an error during configuration, the LED will blink rapidly.

When the FPGA has been successfully configured, the behavior of the LED is application-specific. For example, if a USB keyboard is plugged in, a rapid blink will signal the receipt of an HID input report from the keyboard.

## 3 Memory

The Basys 3 board contains a 32Mbit non-volatile serial Flash device, which is attached to the Artix-7 FPGA using a dedicated quad-mode (x4) SPI bus. The connections and pin assignments between the FPGA and the serial flash device are shown in Fig. 4.

FPGA configuration files can be written to the Quad SPI Flash (Spansion part number S25FL032), and mode settings are available to cause the FPGA to automatically read a configuration from this device at power on. An Artix-7 35T configuration file requires just over two Mbytes of memory, leaving approximately 48% of the flash device available for user data.

**NOTE:** Refer to the manufacturer's data sheets and the reference designs posted on Digilent's website for more information about the memory devices.

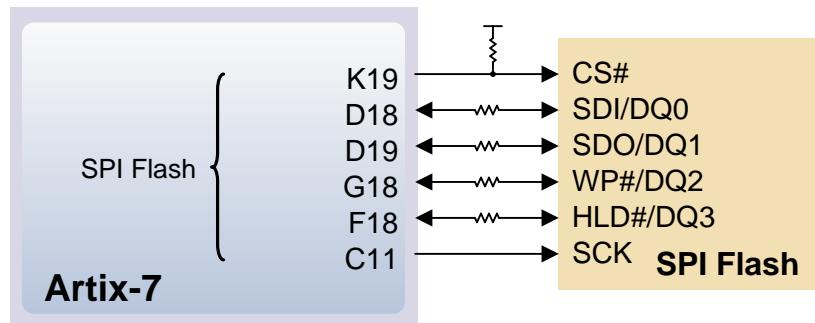


Figure 4. Basys 3 external memory.

## 4 Oscillators/Clocks

The Basys 3 board includes a single 100 MHz oscillator connected to pin W5 (W5 is a MRCC input on bank 34). The input clock can drive MMCMs or PLLs to generate clocks of various frequencies and with known phase relationships that may be needed throughout a design. Some rules restrict which MMCMs and PLLs may be driven by the 100 MHz input clock. For a full description of these rules and of the capabilities of the Artix-7 clocking resources, refer to the "7 Series FPGAs Clocking Resources User Guide" available from Xilinx.

Xilinx offers the LogiCORE™ Clocking Wizard IP to help users generate the different clocks required for a specific design. This wizard properly instantiates the needed MMCMs and PLLs based on the desired frequencies and phase relationships specified by the user. The wizard will then output an easy to use wrapper component around these clocking resources that can be inserted into the user's design. The Clocking Wizard can be accessed from within IP Catalog, which can be found under the Project Manager section of the Flow Navigator in Vivado.

## 5 USB-UART Bridge (Serial Port)

The Basys 3 includes an FTDI FT2232HQ USB-UART bridge (attached to connector J4) that allows you to use PC applications to communicate with the board using standard Windows COM port commands. Free USB-COM port drivers, available from [www.ftdichip.com](http://www.ftdichip.com) under the "Virtual Com Port" or VCP heading, convert USB packets to UART/serial port data. Serial port data is exchanged with the FPGA using a two-wire serial port (TXD/RXD). After the drivers are installed, I/O commands can be used from the PC directed to the COM port to produce serial data traffic on the B18 and A18 FPGA pins.

Two on-board status LEDs provide visual feedback on traffic flowing through the port: the transmit LED (LD18) and the receive LED (LD17). Signal names that imply direction are from the point-of-view of the DTE (Data Terminal Equipment), in this case the PC.

The FT2232HQ is also used as the controller for the Digilent USB-JTAG circuitry, but the USB-UART and USB-JTAG functions behave entirely independent of one another. Programmers interested in using the UART functionality of the FT2232 within their design do not need to worry about the JTAG circuitry interfering with the UART data transfers, and vice-versa. The combination of these two features into a single device allows the Basys 3 to be programmed, communicated with via UART, and powered from a computer attached with a single Micro USB cable. The connections between the FT2232HQ and the Artix-7 are shown in Fig. 6.

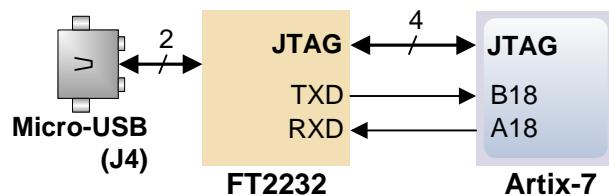


Figure 6. Basys 3 FT2232HQ connections.

## 6 USB HID Host

The Auxiliary Function microcontroller (Microchip PIC24FJ128) provides the Basys 3 with USB HID host capability. After power-up, the microcontroller is in configuration mode, either downloading a bitstream to the FPGA or waiting for it to be programmed from other sources. Once the FPGA is programmed, the microcontroller switches to application mode, which in this case is USB HID Host mode. Firmware in the microcontroller can drive a mouse or a keyboard attached to the type A USB connector at J2 labeled "USB." Hub support is not currently available, so only a single mouse or a single keyboard can be used. The PIC24 drives several signals into the FPGA – two are used to implement a standard PS/2 interface for communication with a mouse or keyboard, and the others are connected to the FPGA's two-wire serial programming port, so the FPGA can be programmed from a file stored on a USB pen drive.

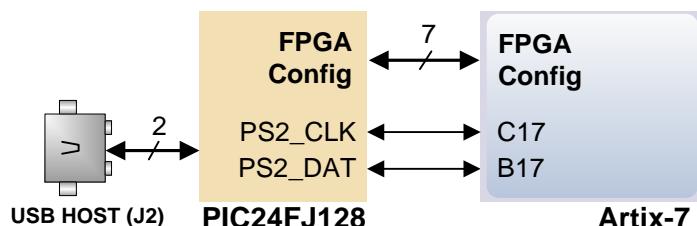


Figure 7. Basys 3 PIC24 connections.

## 6.1 HID Controller

The Auxiliary Function microcontroller hides the USB HID protocol from the FPGA and emulates an old-style PS/2 bus. The microcontroller behaves just like a PS/2 keyboard or mouse would. This means new designs can re-use existing PS/2 IP cores. Mice and keyboards that use the PS/2 protocol use a two-wire serial bus (clock and data) to communicate with a host. On the Basys 3, the microcontroller emulates a PS/2 device while the FPGA plays the role of the host. Both the mouse and the keyboard use 11-bit words that include a start bit, data byte (LSB first), odd parity, and stop bit, but the data packets are organized differently, and the keyboard interface allows bi-directional data transfers (so the host device can illuminate state LEDs on the keyboard). Bus timings are shown in Fig. 8.

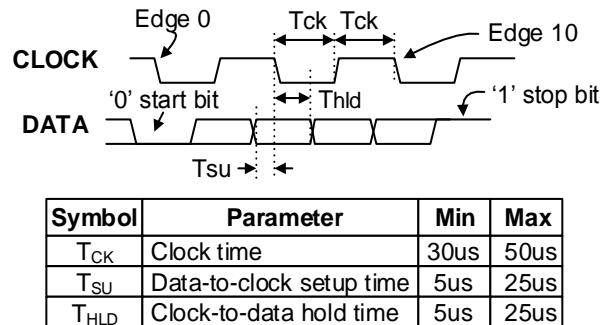


Figure 8. PS/2 device-to host timing diagram.

The clock and data signals are only driven when data transfers occur; otherwise they are held in the idle state at logic '1.' This requires that when the PS/2 signals are used in a design, internal pull-ups must be enabled in the FPGA on the data and clock pins. The clock signal is normally driven by the device, but may be held low by the host in special cases. The timings define signal requirements for mouse-to-host communications and bi-directional keyboard communications. A PS/2 interface circuit can be implemented in the FPGA to create a keyboard or mouse interface.

When a keyboard or mouse is connected to the Basys 3, a "self-test passed" command (0xAA) is sent to the host. After this, commands may be issued to the device. Since both the keyboard and the mouse use the same PS/2 port, one can tell the type of device connected using the device ID. This ID can be read by issuing a Read ID command (0xF2). Also, a mouse sends its ID (0x00) right after the "self-test passed" command, which distinguishes it from a keyboard.

## 6.2 Keyboard

The keyboard uses open-collector drivers so the keyboard, or an attached host device, can drive the two-wire bus (if the host device will not send data to the keyboard, then the host can use input-only ports).

PS/2-style keyboards use scan codes to communicate key press data. Each key is assigned a code that is sent whenever the key is pressed. If the key is held down, the scan code will be sent repeatedly about once every 100ms. When a key is released, an F0 key-up code is sent, followed by the scan code of the released key. If a key can be shifted to produce a new character (like a capital letter), then a shift character is sent in addition to the scan code and the host must determine which ASCII character to use. Some keys, called extended keys, send an E0 ahead of the scan code (and they may send more than one scan code). When an extended key is released, an E0 F0 key-up code is sent, followed by the scan code. Scan codes for most keys are shown in Fig. 9.

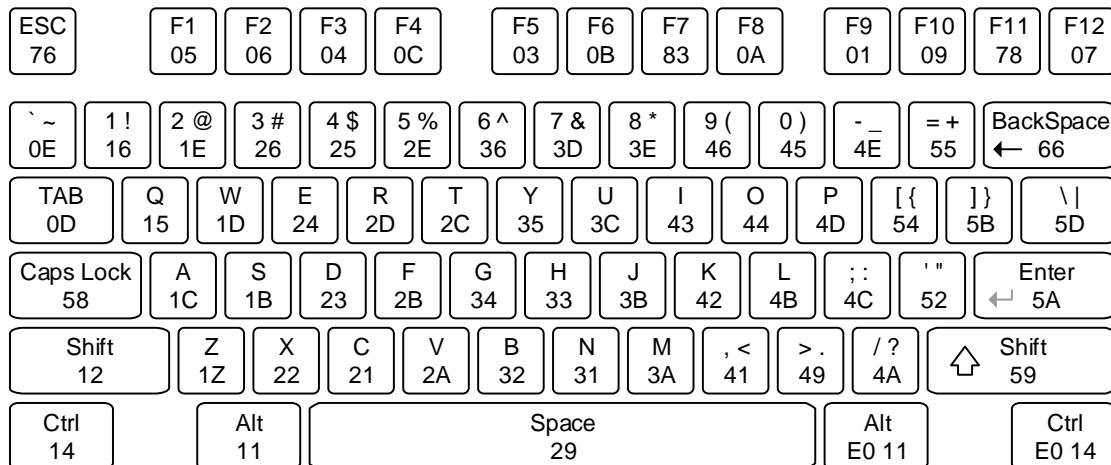


Figure 9. Keyboard scan codes.

A host device can also send data to the keyboard. Table 3 shows a list of some common commands a host might send.

The keyboard can send data to the host only when both the data and clock lines are high (or idle). Because the host is the bus master, the keyboard must check to see whether the host is sending data before driving the bus. To facilitate this, the clock line is used as a "clear to send" signal. If the host drives the clock line low, the keyboard must not send any data until the clock is released. The keyboard sends data to the host in 11-bit words that contain a '0' start bit, followed by 8-bits of scan code (LSB first), followed by an odd parity bit, and terminated with a '1' stop bit. The keyboard generates 11 clock transitions (at 20 to 30 KHz) when the data is sent, and data is valid on the falling edge of the clock.

Command	Action
ED	Set Num Lock, Caps Lock, and Scroll Lock LEDs. Keyboard returns FA after receiving ED, then host sends a byte to set LED status: bit 0 sets Scroll Lock, bit 1 sets Num Lock, and bit 2 sets Caps lock. Bits 3 to 7 are ignored.
EE	Echo (test). Keyboard returns EE after receiving EE.
F3	Set scan code repeat rate. Keyboard returns F3 on receiving FA, then host sends second byte to set the repeat rate.
FE	Resend. FE directs keyboard to re-send most recent scan code.
FF	Reset. Resets the keyboard.

Table 3. Keyboard commands.

## 6.3 Mouse

Once entered in stream mode and data reporting has been enabled, the mouse outputs a clock and data signal when it is moved. Otherwise, these signals remain at logic '1.' Each time the mouse is moved, three 11-bit words are sent from the mouse to the host device, as shown in Fig. 10. Each of the 11-bit words contains a '0' start bit, followed by 8 bits of data (LSB first), followed by an odd parity bit, and terminated with a '1' stop bit. Thus, each data transmission contains 33 bits, where bits 0, 11, and 22 are '0' start bits, and bits 11, 21, and 33 are '1' stop bits. The three 8-bit data fields contain movement data as shown in the Fig. 10. Data is valid at the falling edge of the clock, and the clock period is 20 to 30 KHz.

The mouse assumes a relative coordinate system wherein moving the mouse to the right generates a positive number in the X field, and moving to the left generates a negative number. Likewise, moving the mouse up generates a positive number in the Y field, and moving down represents a negative number (the XS and YS bits in the status byte are the sign bits – a '1' indicates a negative number). The magnitude of the X and Y numbers represent the rate of mouse movement; the larger the number, the faster the mouse is moving (the XV and YV bits in the status byte are movement overflow indicators – a '1' means overflow has occurred). If the mouse moves continuously, the 33-bit transmissions are repeated every 50ms or so. The L and R fields in the status byte indicate Left and Right button presses (a '1' indicates that the button is being pressed).

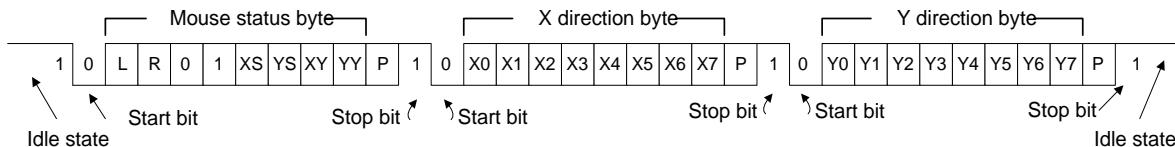


Figure 10. Mouse data format.

The microcontroller also supports Microsoft® IntelliMouse®-type extensions for reporting back a third axis representing the mouse wheel, as shown in Table 4.

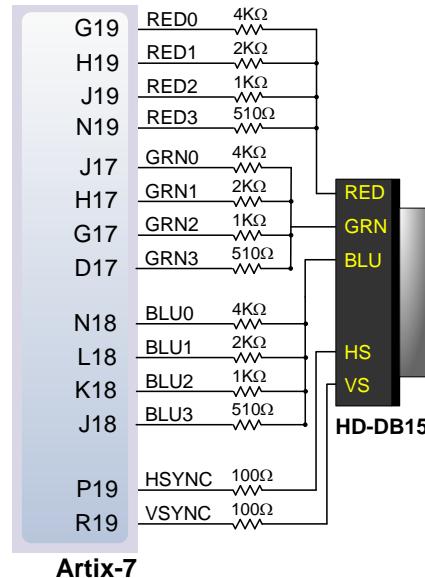
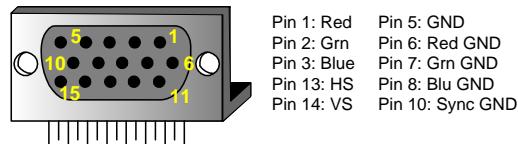
Command	Action
EA	Set stream mode. The mouse responds with "acknowledge" (0xFA) then resets its movement counters and enters stream mode.
F4	Enable data reporting. The mouse responds with "acknowledge" (0xFA) then enables data reporting and resets its movement counters. This command only affects behavior in stream mode. Once issued, mouse movement will automatically generate a data packet.
F5	Disable data reporting. The mouse responds with "acknowledge" (0xFA) then disables data reporting and resets its movement counters.
F3	Set mouse sample rate. The mouse responds with "acknowledge" (0xFA) then reads one more byte from the host. This byte is then saved as the new sample rate, and a new "acknowledge" packet is issued.
FE	Resend. FE directs mouse to re-send last packet.
FF	Reset. The mouse responds with "acknowledge" (0xFA) then enters reset mode.

Table 4. Microsoft Intellimouse-type extensions, commands, and actions.

## 7 VGA Port

**NOTE:** A helpful way to understand the way that VGA signals are transmitted is to understand the method of which CRT (Cathode Ray Tubes) function for displaying images. Although the technology may seem outdated, it is from this legacy that many of the signal names and timings have originated.

The Basys 3 board uses 14 FPGA signals to create a VGA port with 4-bits per color and the two standard sync signals (HS – Horizontal Sync, and VS – Vertical Sync). The color signals use resistor-divider circuits that work in conjunction with the 75 ohm termination resistance of the VGA display to create 16 signal levels each on the red, green, and blue VGA signals. This circuit, shown in Fig. 11, produces video color signals that proceed in equal increments between 0V (fully off) and 0.7V (fully on). Using this circuit, 4096 different colors can be displayed, one for each unique 12-bit pattern. A video controller circuit must be created in the FPGA to drive the sync and color signals with the correct timing in order to produce a working display system.



## 7.1 VGA System Timing

VGA signal timings are specified, published, copyrighted, and sold by the VESA® organization ([www.vesa.org](http://www.vesa.org)). The following VGA system timing information is provided as an example of how a VGA monitor might be driven in 640 by 480 mode.

NOTE: For more precise information, or for information on other VGA frequencies, refer to documentation available at the VESA website.

CRT-based VGA displays use amplitude-modulated moving electron beams (or cathode rays) to display information on a phosphor-coated screen. LCD displays use an array of switches that can impose a voltage across a small amount of liquid crystal, thereby changing light permittivity through the crystal on a pixel-by-pixel basis. Although the following description is limited to CRT displays, LCD displays have evolved to use the same signal timings as CRT displays (so the "signals" discussion below pertains to both CRTs and LCDs). Color CRT displays use three electron beams (one for red, one for blue, and one for green) to energize the phosphor that coats the inner side of the display end of a cathode ray tube (see Fig. 12).

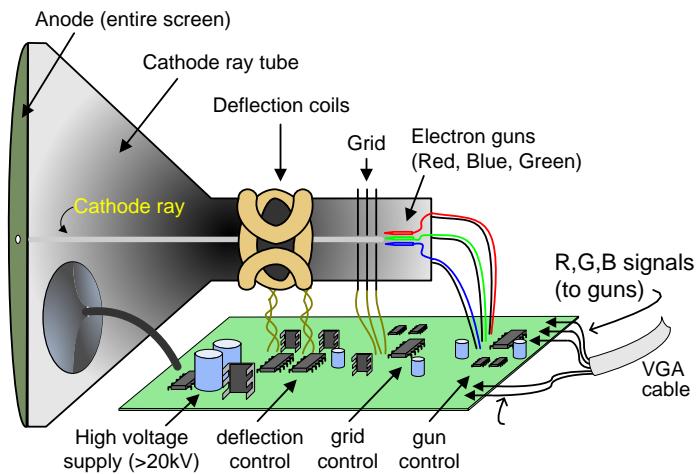


Figure 12. Color CRT display.

Electron beams emanate from "electron guns" which are finely-pointed heated cathodes placed in close proximity to a positively charged annular plate called a "grid." The electrostatic force imposed by the grid pulls rays of energized electrons from the cathodes, and those rays are fed by the current that flows into the cathodes. These particle rays are initially accelerated towards the grid, but they soon fall under the influence of the much larger electrostatic force that results from the entire phosphor-coated display surface of the CRT being charged to 20kV (or more). The rays are focused to a fine beam as they pass through the center of the grids, and then they accelerate to impact on the phosphor-coated display surface. The phosphor surface glows brightly at the impact point, and it continues to glow for several hundred microseconds after the beam is removed. The larger the current fed into the cathode, the brighter the phosphor will glow.

Between the grid and the display surface, the beam passes through the neck of the CRT where two coils of wire produce orthogonal electromagnetic fields. Because cathode rays are composed of charged particles (electrons), they can be deflected by these magnetic fields. Current waveforms are passed through the coils to produce magnetic fields that interact with the cathode rays and cause them to transverse the display surface in a "raster" pattern, horizontally from left to right and vertically from top to bottom, as shown in Fig. 13. As the cathode ray moves over the surface of the display, the current sent to the electron guns can be increased or decreased to change the brightness of the display at the cathode ray impact point.

Information is only displayed when the beam is moving in the "forward" direction (left to right and top to bottom), and not during the time the beam is reset back to the left or top edge of the display. Much of the potential display time is therefore lost in "blanking" periods when the beam is reset and stabilized to begin a new horizontal or vertical display pass. The size of the beams, the frequency at which the beam can be traced across the display, and the frequency at which the electron beam can be modulated determine the display resolution.

Modern VGA displays can accommodate different resolutions, and a VGA controller circuit dictates the resolution by producing timing signals to control the raster patterns. The controller must produce synchronizing pulses at 3.3V (or 5V) to set the frequency at which current flows through the deflection coils, and it must ensure that video data is applied to the electron guns at the correct time. Raster video displays define a number of "rows" that corresponds to the number of horizontal passes the cathode makes over the display area, and a number of "columns" that corresponds to an area on each row that is assigned to one "picture element" or pixel. Typical displays use from 240 to 1200 rows and from 320 to 1600 columns. The overall size of a display and the number of rows and columns determines the size of each pixel.

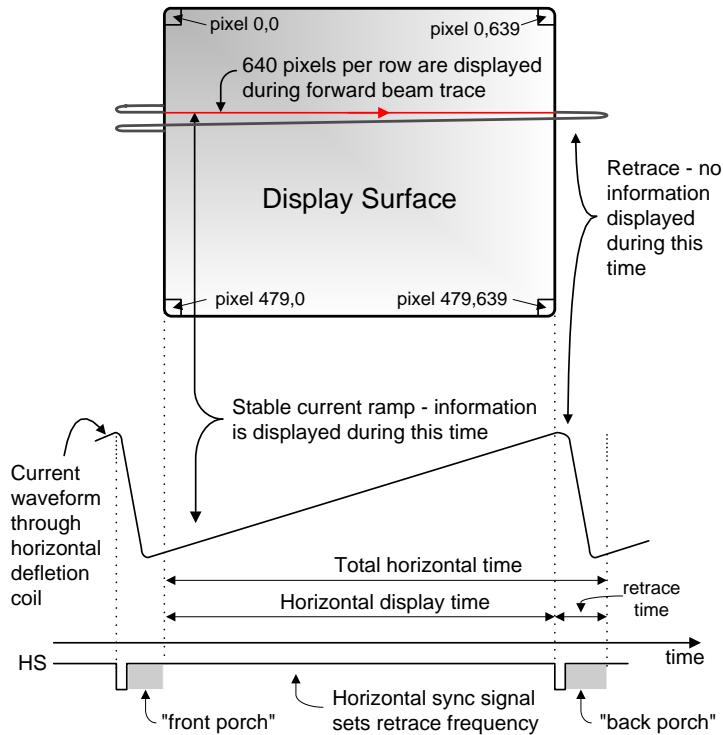
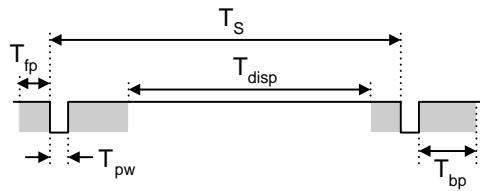


Figure 13. VGA horizontal synchronization.

Video data typically comes from a video refresh memory; with one or more bytes assigned to each pixel location (the Basys 3 uses 12-bits per pixel). The controller must index into video memory as the beams move across the display, and retrieve and apply video data to the display at precisely the time the electron beam is moving across a given pixel.

A VGA controller circuit must generate the HS and VS timings signals and coordinate the delivery of video data based on the pixel clock. The pixel clock defines the time available to display one pixel of information. The VS signal defines the "refresh" frequency of the display, or the frequency at which all information on the display is redrawn. The minimum refresh frequency is a function of the display's phosphor and electron beam intensity, with practical refresh frequencies falling in the 50Hz to 120Hz range. The number of lines to be displayed at a given refresh frequency defines the horizontal "retrace" frequency. For a 640-pixel by 480-row display using a 25 MHz pixel clock and 60 +/-1Hz refresh, the signal timings shown in Fig. 14 can be derived. Timings for sync pulse width and front and back porch intervals (porch intervals are the pre- and post-sync pulse times during which information cannot be displayed) are based on observations taken from actual VGA displays.



Symbol	Parameter	Vertical Sync			Horiz. Sync		
		Time	Clocks	Lines	Time	Clks	
$T_S$	Sync pulse	16.7ms	416,800	521	32 us	800	
$T_{disp}$	Display time	15.36ms	384,000	480	25.6 us	640	
$T_{pw}$	Pulse width	64 us	1,600	2	3.84 us	96	
$T_{fp}$	Front porch	320 us	8,000	10	640 ns	16	
$T_{bp}$	Back porch	928 us	23,200	29	1.92 us	48	

Figure 14. Signal timings for a 640-pixel by 480 row display using a 25 MHz pixel clock and 60 Hz vertical refresh.

A VGA controller circuit, such as the one diagrammed in Fig. 15, decodes the output of a horizontal-sync counter driven by the pixel clock to generate HS signal timings. You can use this counter to locate any pixel location on a given row. Likewise, the output of a vertical-sync counter that increments with each HS pulse can be used to generate VS signal timings, and you can use this counter to locate any given row. These two continually running counters can be used to form an address into video RAM. No time relationship between the onset of the HS pulse and the onset of the VS pulse is specified, so you can arrange the counters to easily form video RAM addresses, or to minimize decoding logic for sync pulse generation.

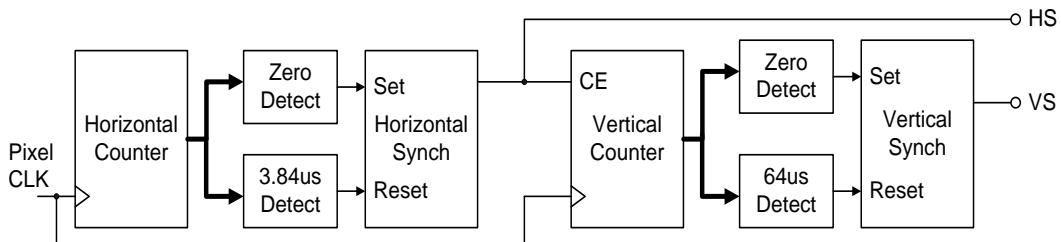


Figure 15. VGA display controller block diagram.

## 8 Basic I/O

The Basys 3 board includes sixteen slide switches, five push buttons, sixteen individual LEDs, and a four-digit seven-segment display, as shown in Fig. 16. The pushbuttons and slide switches are connected to the FPGA via series resistors to prevent damage from inadvertent short circuits (a short circuit could occur if an FPGA pin assigned to a pushbutton or slide switch was inadvertently defined as an output). The five pushbuttons, arranged in a plus-sign configuration, are "momentary" switches that normally generate a low output when they are at rest, and a high output only when they are pressed. Slide switches generate constant high or low inputs depending on their position.

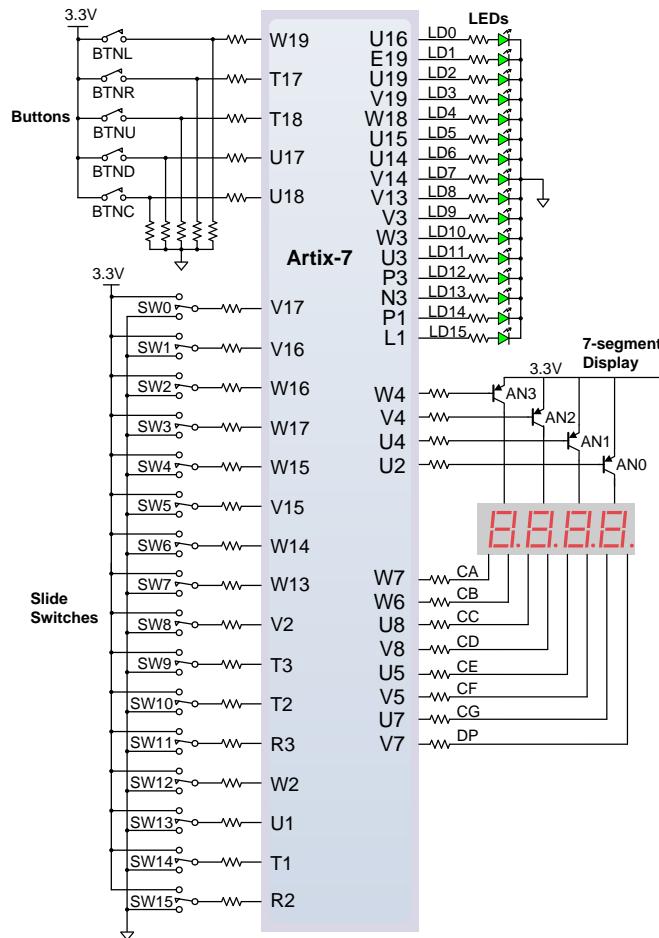


Figure 16. General purpose I/O devices on the Basys 3.

The sixteen individual high-efficiency LEDs are anode-connected to the FPGA via 330 ohm resistors, so they will turn on when a logic high voltage is applied to their respective I/O pin. Additional LEDs, which are not user-accessible, indicate power-on, FPGA programming status, and USB port status.

## 8.1 Seven-Segment Display

The Basys 3 board contains one four-digit common anode seven-segment LED display. Each of the four digits is composed of seven segments arranged in a "figure 8" pattern, with an LED embedded in each segment. Segment LEDs can be individually illuminated, so any one of 128 patterns can be displayed on a digit by illuminating certain LED segments and leaving the others dark, as shown in Fig. 17. Of these 128 possible patterns, the ten corresponding to the decimal digits are the most useful.

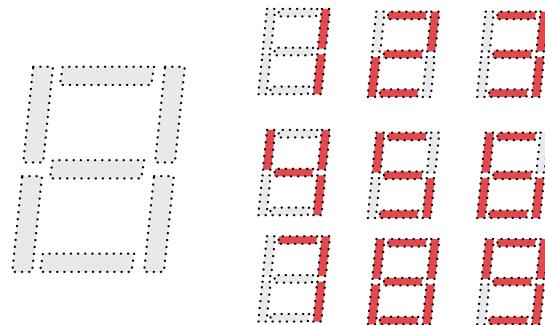


Figure 17. An un-illuminated seven-segment display and nine illumination patterns corresponding to decimal digits.

The anodes of the seven LEDs forming each digit are tied together into one "common anode" circuit node, but the LED cathodes remain separate, as shown in Fig. 18. The common anode signals are available as four "digit enable" input signals to the 4-digit display. The cathodes of similar segments on all four displays are connected into seven circuit nodes labeled CA through CG (for example, the four "D" cathodes from the four digits are grouped together into a single circuit node called "CD"). These seven cathode signals are available as inputs to the 4-digit display. This signal connection scheme creates a multiplexed display, where the cathode signals are common to all digits but they can only illuminate the segments of the digit whose corresponding anode signal is asserted.

To illuminate a segment, the anode should be driven high while the cathode is driven low. However, since the Basys 3 uses transistors to drive enough current into the common anode point, the anode enables are inverted. Therefore, both the AN0..3 and the CA..G/DP signals are driven low when active.

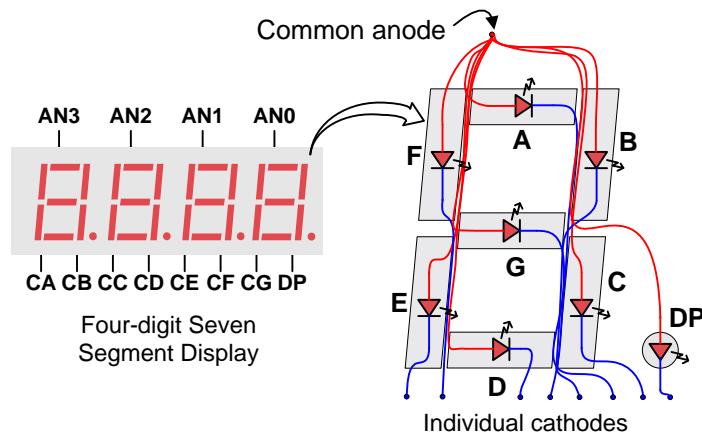


Figure 18. Common anode circuit node.

A scanning display controller circuit can be used to show a four-digit number on this display. This circuit drives the anode signals and corresponding cathode patterns of each digit in a repeating, continuous succession at an update rate that is faster than the human eye can detect. Each digit is illuminated just one-fourth of the time, but because the eye cannot perceive the darkening of a digit before it is illuminated again, the digit appears continuously illuminated. If the update, or "refresh", rate is slowed to around 45Hz, a flicker can be noticed in the display.

For each of the four digits to appear bright and continuously illuminated, all four digits should be driven once every 1 to 16ms, for a refresh frequency of about 1 KHz to 60Hz. For example, in a 62.5Hz refresh scheme, the entire display would be refreshed once every 16ms, and each digit would be illuminated for 1/4 of the refresh cycle, or 4ms. The controller must drive the cathodes low with the correct pattern when the corresponding anode signal is driven high. To illustrate the process, if AN0 is asserted while CB and CC are asserted, then a "1" will be displayed

in digit position 1. Then, if AN1 is asserted while CA, CB, and CC are asserted, a "7" will be displayed in digit position 2. If AN0, CB, and CC are driven for 4ms, and then AN1, CA, CB, and CC are driven for 4ms in an endless succession, the display will show "71" in the first two digits. An example timing diagram for a four-digit controller is shown in Fig. 19.

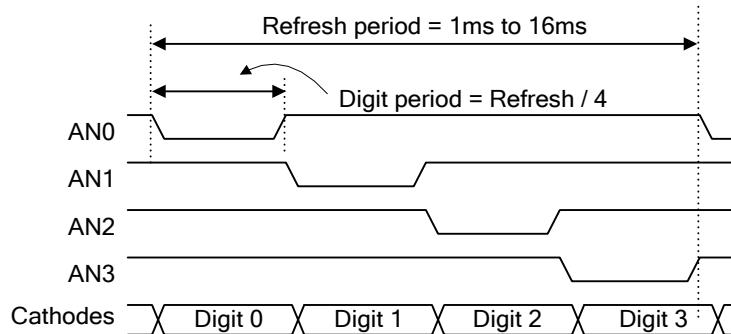


Figure 19. Four digit scanning display controller timing diagram.

## 9 Pmod Ports

The Pmod ports are arranged in a 2x6 right-angle, and are 100-mil female connectors that mate with standard 2x6 pin headers. Each 12-pin Pmod port provides two 3.3V VCC signals (pins 6 and 12), two Ground signals (pins 5 and 11), and eight logic signals, as shown in Fig. 20. The VCC and Ground pins can deliver up to 1A of current. Pmod data signals are not matched pairs, and they are routed using best-available tracks without impedance control or delay matching. Pin assignments for the Pmod I/O connected to the FPGA are shown in Table 6.

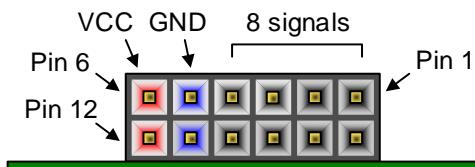


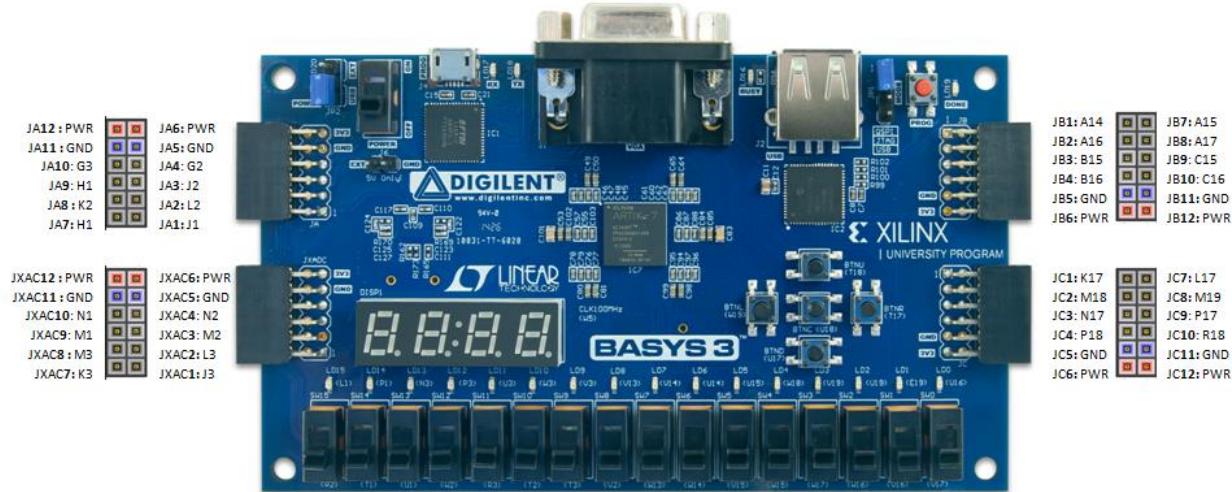
Figure 20. Pmod ports; front view as loaded on PCB.

Pmod JA	Pmod JB	Pmod JC	Pmod XDAC
JA1: J1	JB1: A14	JC1: K17	JXADC1: J3
JA2: L2	JB2: A16	JC2: M18	JXADC2: L3
JA3: J2	JB3: B15	JC3: N17	JXADC3: M2
JA4: G2	JB4: B16	JC4: P18	JXADC4: N2
JA7: H1	JB7: A15	JC7: L17	JXADC7: K3
JA8: K2	JB8: A17	JC8: M19	JXADC8: M3
JA9: H2	JB9: C15	JC9: P17	JXADC9: M1
JA10: G3	JB10: C16	JC10: R18	JXADC10: N1

Table 6. Basys 3 Pmod pin assignment.

Digilent produces a large collection of Pmod accessory boards that can attach to the Pmod expansion ports to add ready-made functions like A/Ds, D/A, motor drivers, sensors, and other functions. See [www.digilentinc.com](http://www.digilentinc.com) for more information.

**Basys3:** Pmod Pin-Out Diagram



## 9.1 Dual Analog/Digital Pmod

The on-board Pmod expansion port, labeled "JXADC", is wired to the auxiliary analog input pins of the FPGA. Depending on the configuration, this connector can be used to input differential analog signals to the analog-to-digital converter inside the Artix-7 (XADC). Any or all pairs in the connector can be configured either as analog input or digital input-output.

The Dual Analog/Digital Pmod on the Basys 3 differs from the rest in the routing of its traces. The eight data signals are grouped into four pairs, with the pairs routed closely coupled for better analog noise immunity. Furthermore, each pair has a partially loaded anti-alias filter laid out on the PCB. The filter does not have capacitors C33-C36. In designs where such filters are desired, the capacitors can be manually loaded by the user.

NOTE: The coupled routing and the anti-alias filters might limit the data speeds when used for digital signals.

The XADC core within the Artix-7 is a dual channel 12-bit analog-to-digital converter capable of operating at 1 MSPS. Either channel can be driven by any of the auxiliary analog input pairs connected to the JXADC header. The XADC core is controlled and accessed from a user design via the Dynamic Reconfiguration Port (DRP). The DRP also provides access to voltage monitors that are present on each of the FPGA's power rails, and a temperature sensor that is internal to the FPGA. For more information on using the XADC core, refer to the Xilinx document titled "7 Series FPGAs and Zynq-7000 All Programmable SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter."

## 10 Built-In Self-Test

A demonstration configuration is loaded into the SPI Flash device on the Basys 3 board during manufacturing. The source code and prebuilt bitstream for this design are available for download from the Digilent website. If the demo configuration is present in the SPI Flash device and the Basys 3 board is powered on in SPI mode, the demo project will allow basic hardware verification. Here is an overview of how this demo drives the different onboard components:

- The user LEDs are illuminated when the corresponding user switch is placed in the on position.
- The VGA port displays feedback from a USB Mouse.
- Connecting a mouse to the USB-HID Mouse port will allow the pointer on the VGA display to be controlled.
- On power-up, each digit of the seven-segment display will display a counter output from 0-9 that increments once a second.
- Pressing BTNU, BTNL, BTNR, or BTND will cause a digit of the seven-segment display to go blank.
- Pressing BTNC will reset the design.
- On power-up, a welcome message is sent over the UART. Also, every time a button is pressed a message is sent. The UART can be connected to using a terminal program with 9600 Baud, 8 data bits, 1 stop bit, and no parity.

All Basys 3 boards are 100% tested during the manufacturing process. If any device on the Basys 3 board fails test or is not responding properly, it is likely that damage occurred during transport or during use. Typical damage includes stressed solder joints and contaminants in switches and buttons resulting in intermittent failures. Stressed solder joints can be repaired by reheating and reflowing solder and contaminants can be cleaned with off-the-shelf electronics cleaning products. If a board fails test within the warranty period, it will be replaced at no cost. Contact Digilent for more details.

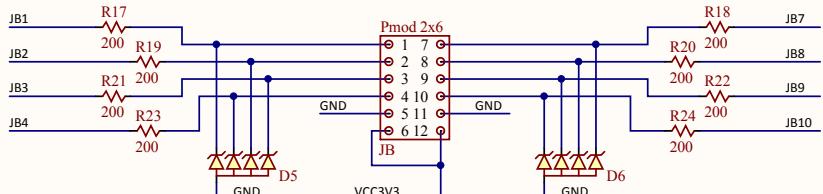
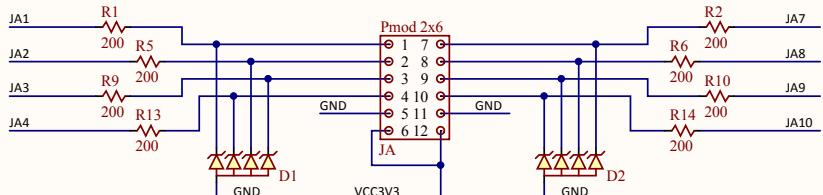
1

2

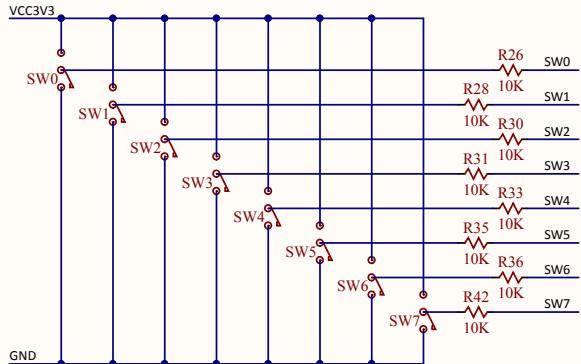
3

4

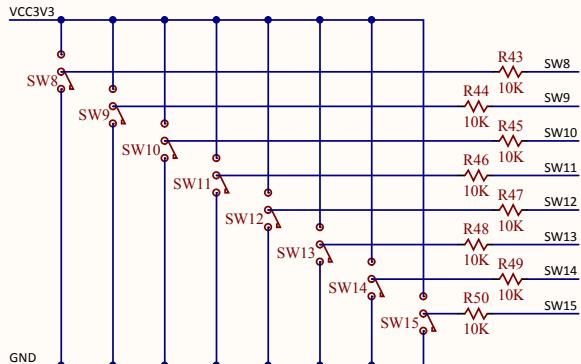
A



B



C



Digilent Inc.

CE

ROHS

Chinese ROHS

Basys 3

XUP

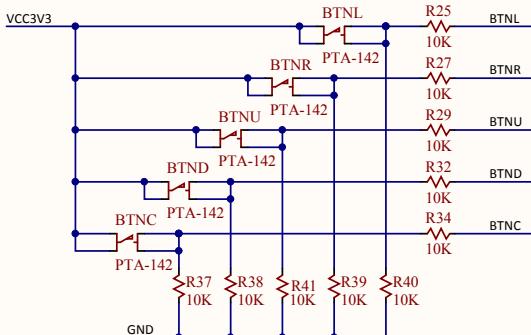
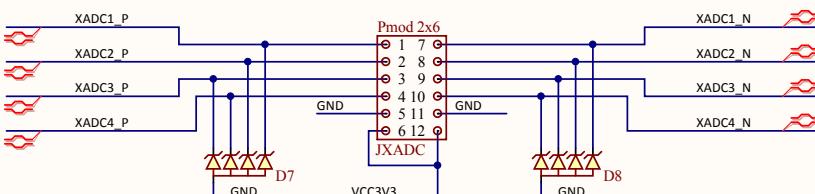
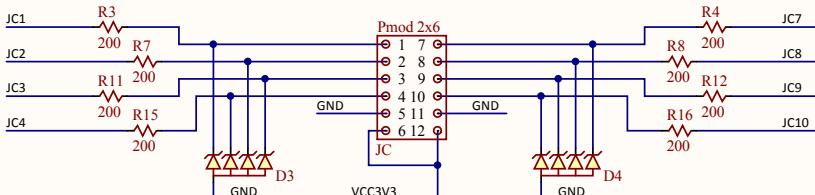
Linear Tech

1

2

3

4



F1  Foot  
 F2  Foot  
 F3  Foot  
 F4  Foot

Title

**Basys 3**

Rev

C.0  
Copyright 2014

Circuit

PMOD, I/O

Doc# 500-183

Engineer MTA

Author GMA

Date 5/21/2014

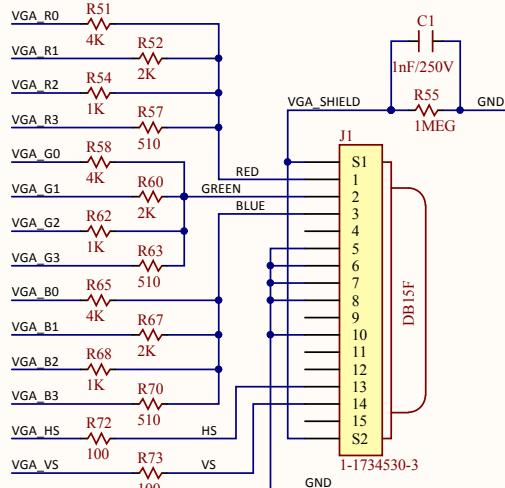
Sheet# 1 out of 8



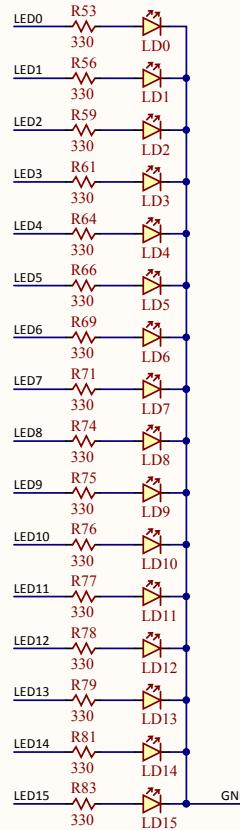
3

4

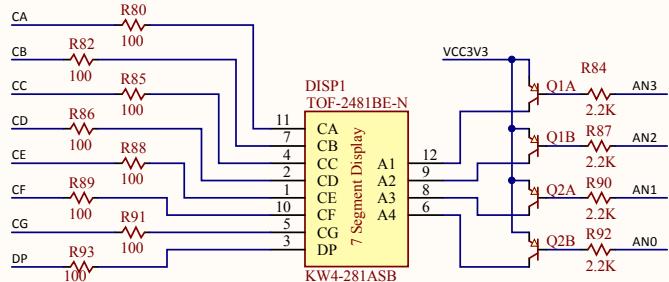
A



B



C



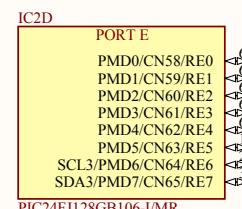
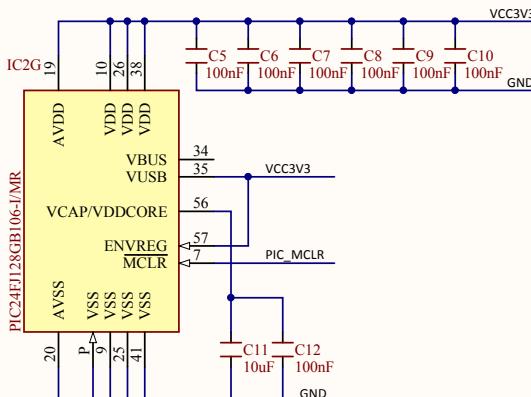
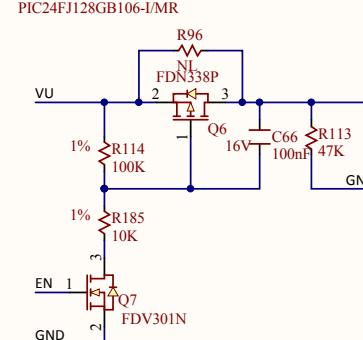
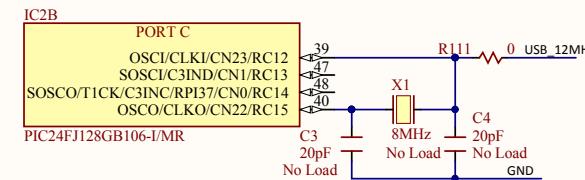
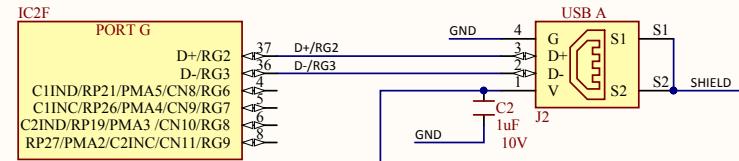
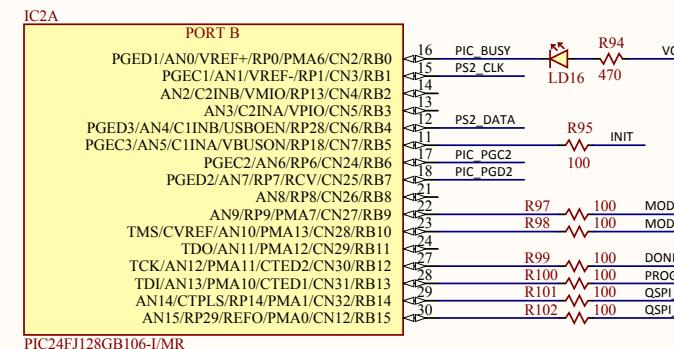
D

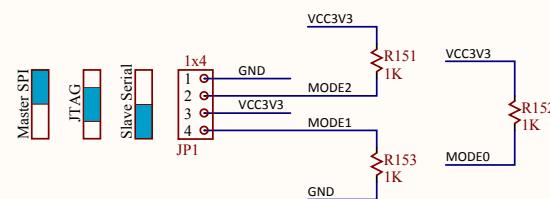
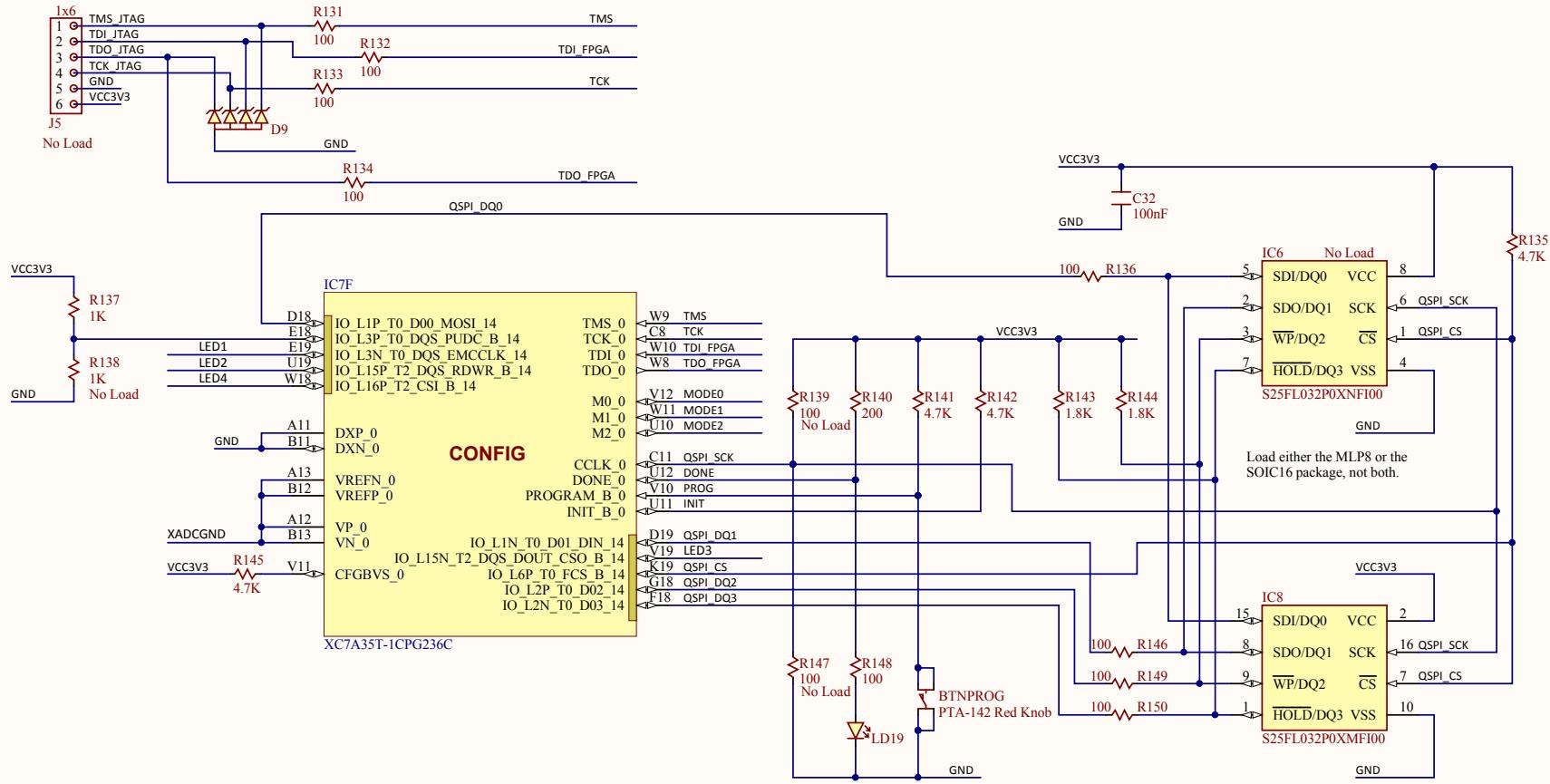
Title  
**Basys 3**

Rev  
**C.0**  
Copyright 2014

Circuit 7 SEG, VGA, LEDs
Doc# 500-183
Engineer MTA
Author GMA
Date 5/21/2014
Sheet# 2 out of 8







Title

## Circuit CONFIG, SPI FLASH

Doc# 500-183

500-TC

Author GMA

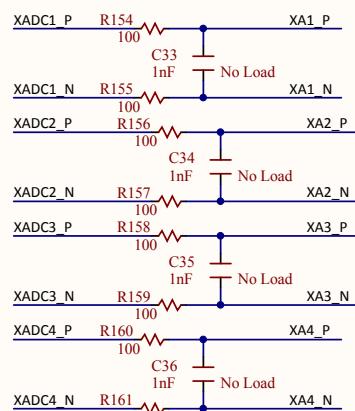
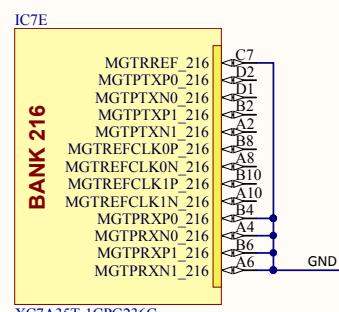
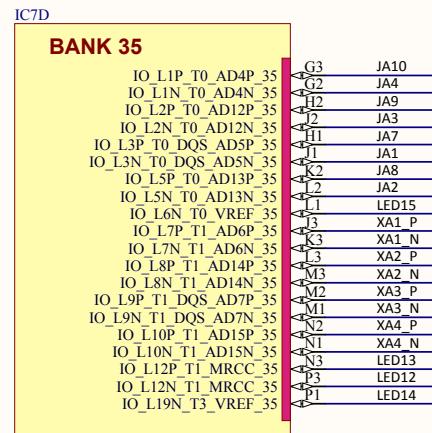
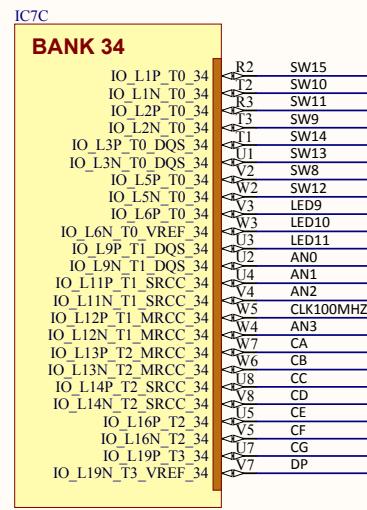
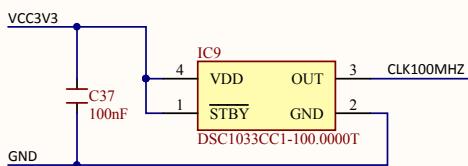
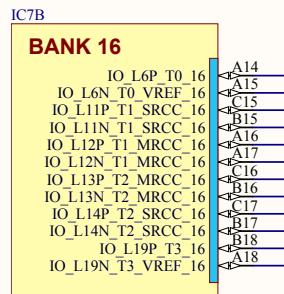
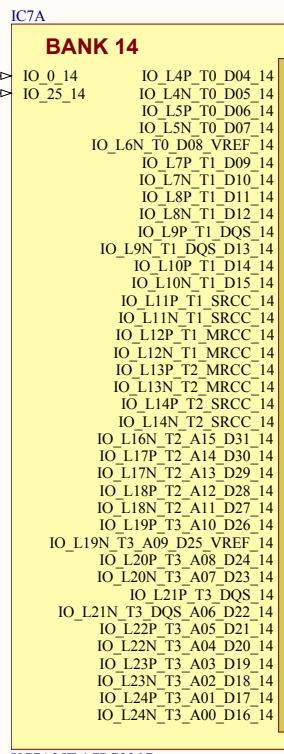
5/21/2015

Sheet# 5 out

Rev  
C.0  
Copyright 2014

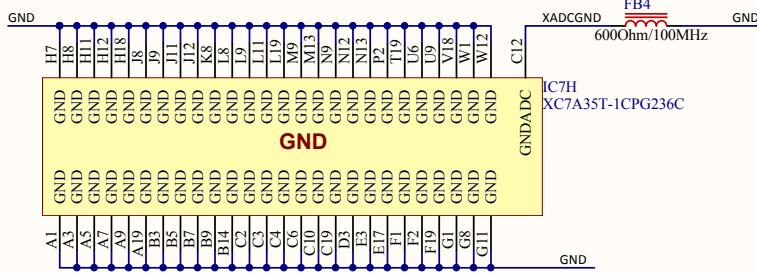
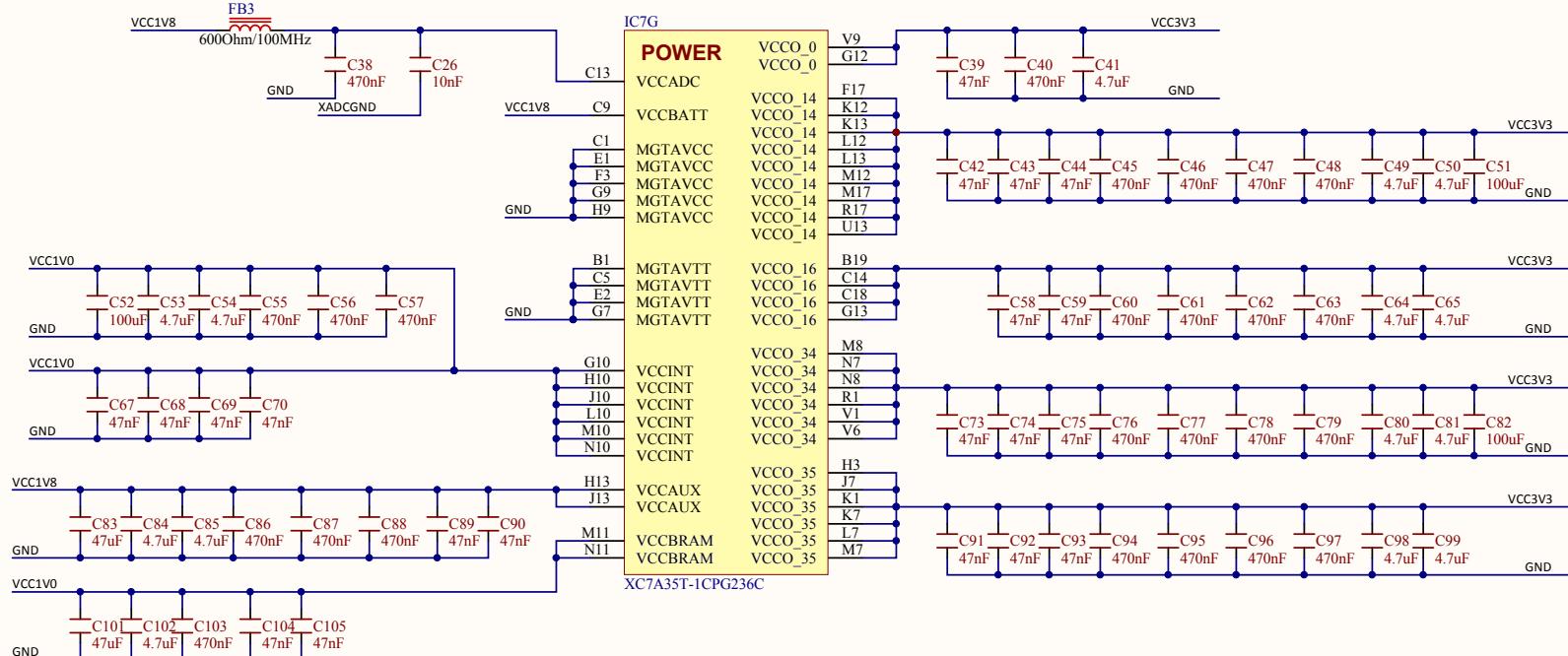
Copyright 2014





Title	Basys 3
Circuit	FPGA BANKS
Doc#	500-183
Engineer	MTA
Author	GMA
Date	5/21/2014
Sheet#	6 out of 8





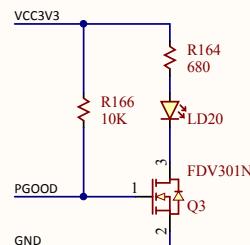
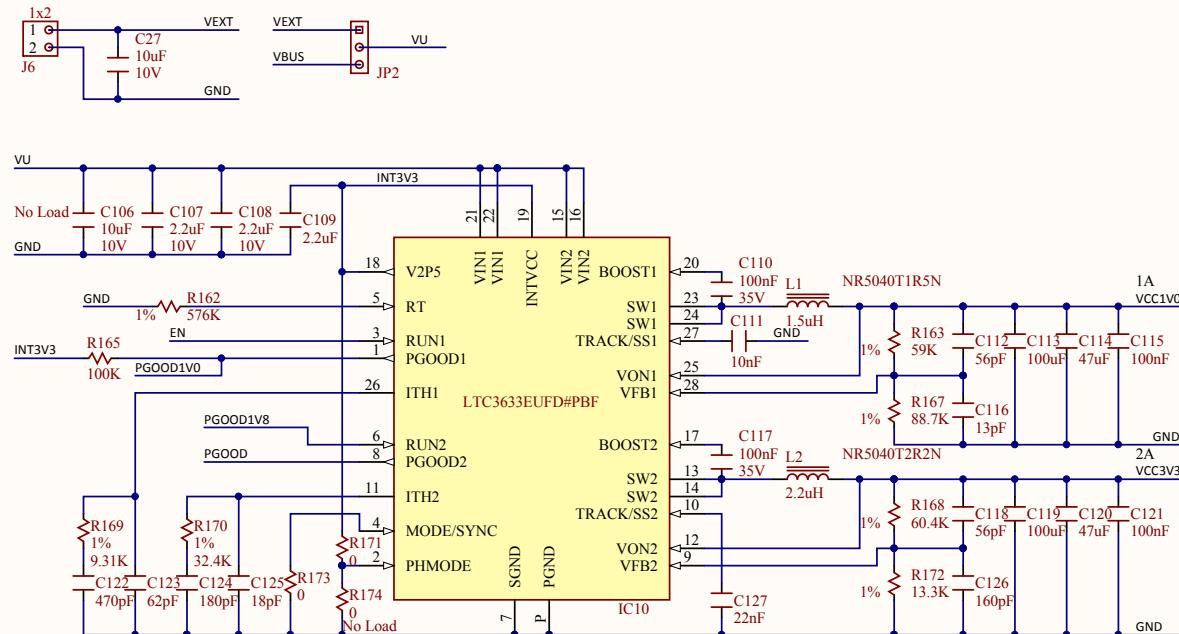
Title  
**Basys 3**

Rev  
**C.0**  
Copyright 2014

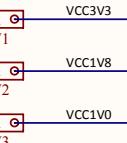
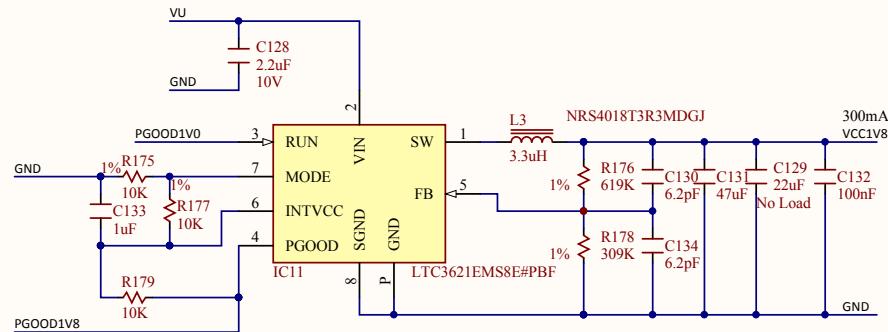
Circuit	FPGA POWER
Doc#	500-183
Engineer	MTA
Author	GMA
Date	5/21/2014
Sheet#	7 out of 8



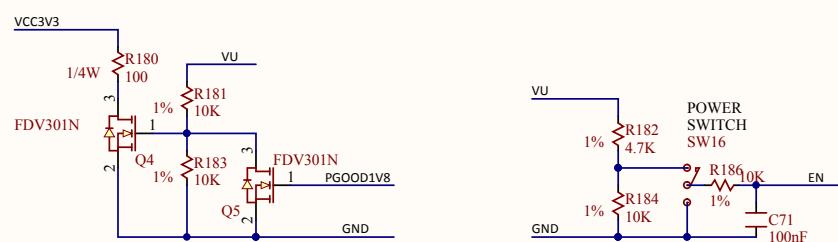
A



B



D



Title  
**Basys 3**  
Rev  
**C.0**  
Copyright 2014

Circuit POWER
Doc# 500-183
Engineer MTA
Author GMA
Date 5/21/2014
Sheet# 8 out of 8





# ESP8266EX Datasheet

**Version 4.3**

Espressif Systems IOT Team

<http://bbs.espressif.com/>

Copyright © 2015



## Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice. THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

The Wi-Fi Alliance Member Logo is a trademark of the WiFi Alliance.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.

Copyright © 2015 Espressif Systems. All rights reserved.



# Table of Contents

1.	General Overview .....	6
1.1.	Introduction .....	6
1.2.	Features .....	7
1.3.	Parameters .....	7
1.4.	Ultra Low Power Technology .....	9
1.5.	Major Applications.....	9
2.	Hardware Overview.....	11
2.1.	Pin Definitions .....	11
2.2.	Electrical Characteristics.....	13
2.3.	Power Consumption.....	13
2.4.	Receiver Sensitivity.....	14
2.5.	MCU.....	15
2.6.	Memory Organization.....	15
2.6.1.	Internal SRAM and ROM.....	15
2.6.2.	External SPI Flash.....	15
2.7.	AHB and AHB Blocks.....	16
3.	Pins and Definitions .....	17
3.1.	GPIO .....	17
3.1.1.	General Purpose Input/Output Interface (GPIO) .....	17



3.2.	Secure Digital Input/Output Interface (SDIO) .....	18
3.3.	Serial Peripheral Interface (SPI/HSPI).....	18
3.3.1.	General SPI (Master/Slave).....	18
3.3.2.	SDIO / SPI (Slave).....	19
3.3.3.	HSPI (Master/Slave).....	19
3.4.	Inter-integrated Circuit Interface (I2C).....	19
3.5.	I2S .....	20
3.6.	Universal Asynchronous Receiver Transmitter (UART).....	20
3.7.	Pulse-Width Modulation (PWM) .....	21
3.8.	IR Remote Control .....	22
3.9.	ADC (Analog-to-digital Converter) .....	22
3.10.	LED Light and Button .....	24
4.	Firmware & Software Development Kit.....	26
4.1.	Features.....	26
5.	Power Management .....	27
6.	Clock Management .....	28
6.1.	High Frequency Clock.....	28
6.2.	External Reference Requirements .....	29
7.	Radio.....	29
7.1.	Channel Frequencies .....	30
7.2.	2.4 GHz Receiver .....	30
7.3.	2.4 GHz Transmitter .....	30



7.4. Clock Generator.....	30
8. Appendix: QFN32 Package Size .....	31



## 1. General Overview

### 1.1. Introduction

Espressif Systems' Smart Connectivity Platform (ESCP) is a set of high performance, high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed WiFi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

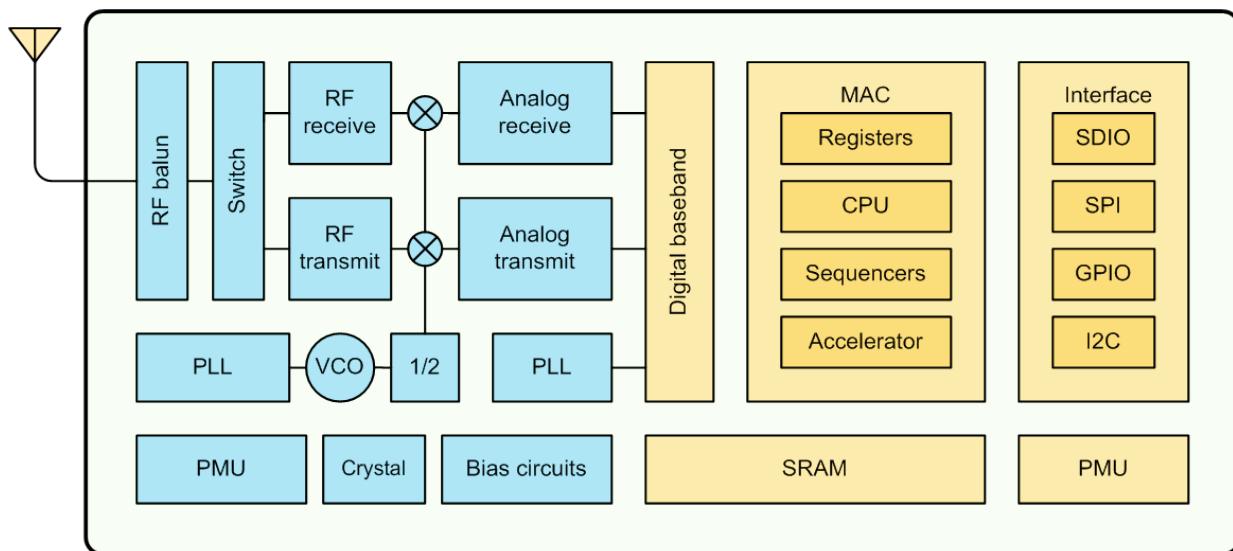


Figure 1 ESP8266EX Block Diagram

ESP8266EX offers a complete and self-contained WiFi networking solution; it can be used to host the application or to offload WiFi networking functions from another application processor.

When ESP8266EX hosts the application, it boots up directly from an external flash. It has integrated cache to improve the performance of the system in such applications.

Alternately, serving as a WiFi adapter, wireless internet access can be added to any micro controller-based design with simple connectivity (SPI/SDIO or I2C/UART interface).

ESP8266EX is among the most integrated WiFi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the WiFi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; sample codes for such applications are provided in the software development kit (SDK).



Espressif Systems' Smart Connectivity Platform (ESCP) demonstrates sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

### 1.2. Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Deep sleep power <10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C
- FCC, CE, TELEC, WiFi Alliance, and SRRC certified

### 1.3. Parameters

Table 1 Parameters



Categories	Items	Values
WiFi Parameters	Certificates	FCC/CE/TELEC/SRRC
	WiFi Protocols	802.11 b/g/n
	Frequency Range	2.4G-2.5G (2400M-2483.5M)
	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)
	Types of Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware Parameters	Peripheral Bus	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	5x5mm
	External Interface	N/A
Software Parameters	WiFi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP



	User Configuration	AT Instruction Set, Cloud Server, Android/ iOS App
--	--------------------	---

## 1.4. Ultra Low Power Technology

ESP8266EX has been designed for mobile, wearable electronics and Internet of Things applications with the aim of achieving the lowest power consumption with a combination of several proprietary techniques. The power saving architecture operates mainly in 3 modes: active mode, sleep mode and deep sleep mode.

By using advance power management techniques and logic to power-down functions not required and to control switching between sleep and active modes, ESP8266EX consumes about than 60uA in deep sleep mode (with RTC clock still running) and less than 1.0mA (DTIM=3) or less than 0.5mA (DTIM=10) to stay connected to the access point.

When in sleep mode, only the calibrated real-time clock and watchdog remains active. The real-time clock can be programmed to wake up the ESP8266EX at any required interval.

The ESP8266EX can be programmed to wake up when a specified condition is detected. This minimal wake-up time feature of the ESP8266EX can be utilized by mobile device SOCs, allowing them to remain in the low-power standby mode until WiFi is needed.

In order to satisfy the power demand of mobile and wearable electronics, ESP8266EX can be programmed to reduce the output power of the PA to fit various application profiles, by trading off range for power consumption.

## 1.5. Major Applications

Major fields of ESP8266EX applications to Internet-of-Things include:

- Home Appliances
- Home Automation
- Smart Plug and lights
- Mesh Network
- Industrial Wireless Control
- Baby Monitors
- IP Cameras
- Sensor Networks
- Wearable Electronics



- WiFi Location-aware Devices
- Security ID Tags
- WiFi Position System Beacons



## 2. Hardware Overview

### 2.1. Pin Definitions

The pin assignments for 32-pin QFN package is illustrated in Fig.2.

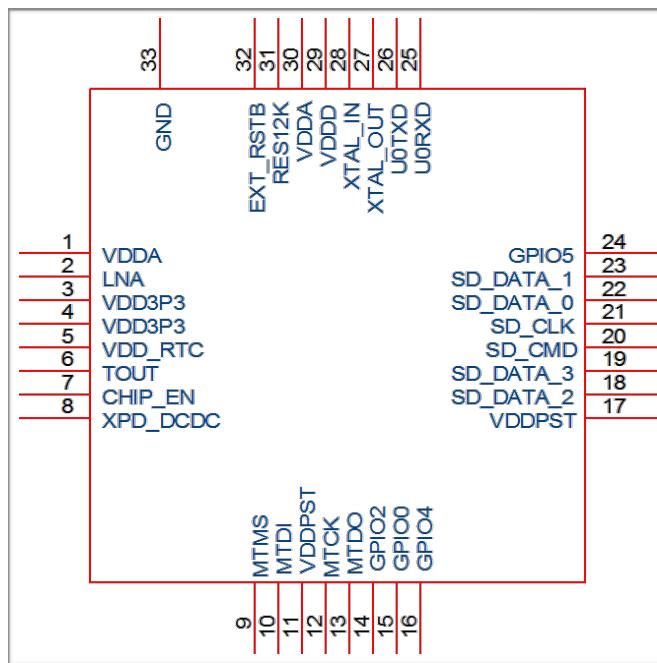


Figure 2 Pin Assignments

Table 2 below presents an overview on the general pin attributes and the functions of each pin.

Table 2 Pin Definitions

Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.0 ~3.6V
2	LNA	I/O	RF Antenna Interface. Chip Output Impedance=50Ω No matching required but we recommend that the n-type matching network is retained.
3	VDD3P3	P	Amplifier Power 3.0~3.6V
4	VDD3P3	P	Amplifier Power 3.0~3.6V
5	VDD_RTC	P	NC (1.1V)



6	TOUT	I	ADC Pin (note: an internal pin of the chip) can be used to check the power voltage of VDD3P3 (Pin 3 and Pin4) or the input voltage of TOUT (Pin 6). These two functions cannot be used simultaneously.
7	CHIP_EN	I	Chip Enable. High: On, chip works properly; Low: Off, small current
8	XPD_DCDC	I/O	Deep-Sleep Wakeup; GPIO16
9	MTMS	I/O	GPIO14; HSPI_CLK
10	MTDI	I/O	GPIO12; HSPI_MISO
11	VDDPST	P	Digital/IO Power Supply (1.8V~3.3V)
12	MTCK	I/O	GPIO13; HSPI_MOSI; UART0_CTS
13	MTDO	I/O	GPIO15; HSPI_CS; UART0_RTS
14	GPIO2	I/O	UART Tx during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPI_CS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/IO Power Supply (1.8V~3.3V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R: 200Ω); SPIHD; HSPIHD; GPIO9
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R: 200Ω); SPIWP; HSPIWP; GPIO10
20	SDIO_CMD	I/O	Connect to SD_CMD (Series R: 200Ω); SPI_CS0; GPIO11
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R: 200Ω); SPI_CLK; GPIO6
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R: 200Ω); SPI_MSIO; GPIO7
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R: 200Ω); SPI_MOSI; GPIO8
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART Tx during flash programming; GPIO1; SPI_CS1
27	XTAL_OUT	I/O	Connect to crystal oscillator output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal oscillator input
29	VDDD	P	Analog Power 3.0V~3.6V
30	VDDA	P	Analog Power 3.0V~3.6V
31	RES12K	I	Serial connection with a 12 kΩ resistor and connect to the ground
32	EXT_RSTB	I	External reset signal (Low voltage level: Active)



**Note:** GPIO2, GPIO0, MTDO can be configurable as 3-bit SDIO mode.

## 2.2. Electrical Characteristics

Table 3 ESP8266EX Electrical Characteristics

Parameters	Conditions	Min	Typical	Max	Unit
Storage Temperature Range		-40	Normal	125	°C
Maximum Soldering Temperature	IPC/JEDEC J-STD-020			260	°C
Working Voltage Value		3.0	3.3	3.6	V
I/O	V <sub>IL</sub> /V <sub>IH</sub>	-0.3/0.75V <sub>IO</sub>	0.25V <sub>IO</sub> /3.6	0.1V <sub>IO</sub> /N	V
	V <sub>OL</sub> /V <sub>OH</sub>				
	I <sub>MAX</sub>			12	mA
Electrostatic Discharge (HBM)	TAMB=25°C			2	kV
Electrostatic Discharge (CDM)	TAMB=25°C			0.5	kV

## 2.3. Power Consumption

The following current consumption is based on 3.3V supply, and 25°C ambient, using internal regulators. Measurements are done at antenna port without SAW filter. All the transmitter's measurements are based on 90% duty cycle, continuous transmit mode.

Table 4 Description on Power Consumption

Parameters	Min	Typical	Max	Unit
Tx802.11b, CCK 11Mbps, P OUT=+17dBm		170		mA
Tx 802.11g, OFDM 54Mbps, P OUT =+15dBm		140		mA
Tx 802.11n, MCS7, P OUT =+13dBm		120		mA
Rx 802.11b, 1024 bytes packet length , -80dBm		50		mA
Rx 802.11g, 1024 bytes packet length, -70dBm		56		mA
Rx 802.11n, 1024 bytes packet length, -65dBm		56		mA
Modem-Sleep①		15		mA
Light-Sleep②		0.9		mA
Deep-Sleep③		10		uA
Power Off		0.5		uA



①: Modem-Sleep requires the CPU to be working, as in PWM or I2S applications. According to 802.11 standards (like U-APSD), it saves power to shut down the WiFi Modem circuit while maintaining a WiFi connection with no data transmission. E.g. in DTIM3, to maintain a sleep 300ms-wake 3ms cycle to receive AP's Beacon packages, the current is about 15mA

②: During Light-Sleep, the CPU may be suspended in applications like WiFi switch. Without data transmission, the WiFi Modem circuit can be turned off and CPU suspended to save power according to the 802.11 standard (U-APSD). E.g. in DTIM3, to maintain a sleep 300ms-wake 3ms cycle to receive AP's Beacon packages, the current is about 0.9mA.

③: Deep-Sleep does not require WiFi connection to be maintained. For application with long time lags between data transmission, e.g. a temperature sensor that checks the temperature every 100s, sleep 300s and waking up to connect to the AP (taking about 0.3~1s), the overall average current is less than 1mA.

## 2.4. Receiver Sensitivity

The following are measured under room temperature conditions with 3.3V and 1.1V power supplies.

Table 5 Receiver Sensitivity

Parameters	Min	Typical	Max	Unit
Input frequency	2412		2484	MHz
Input impedance		50		$\Omega$
Input reflection			-10	dB
Output power of PA for 72.2Mbps	15.5	16.5	17.5	dBm
Output power of PA for 11b mode	19.5	20.5	21.5	dBm
Sensitivity				
DSSS, 1Mbps		-98		dBm
CCK, 11Mbps		-91		dBm
6Mbps (1/2 BPSK)		-93		dBm
54Mbps (3/4 64-QAM)		-75		dBm
HT20, MCS7 (65Mbps, 72.2Mbps)		-72		dBm
<b>Adjacent Channel Rejection</b>				
OFDM, 6Mbps		37		dB
OFDM, 54Mbps		21		dB
HT20, MCS0		37		dB
HT20, MCS7		20		dB



## 2.5. MCU

ESP8266EX is embedded with Tensilica L106 32-bit micro controller (MCU), which features extra low power consumption and 16-bit RSIC. The CPU clock speed is 80MHz. It can also reach a maximum value of 160MHz. Real Time Operation System (RTOS) is enabled. Currently, only 20% of MIPS has been occupied by the WiFi stack, the rest can all be used for user application programming and development. The following interfaces can be used to connect to the MCU embedded in ESP8266EX:

- Programmable RAM/ROM interfaces (iBus), which can be connected with memory controller, and can also be used to visit external flash;
- Data RAM interface (dBus), which can be connected with memory controller;
- AHB interface, can be used to visit the register.

## 2.6. Memory Organization

### 2.6.1. Internal SRAM and ROM

ESP8266EX WiFi SoC is embedded with memory controller, including SRAM and ROM. MCU can visit the memory units through iBus, dBus, and AHB interfaces. All memory units can be visited upon request, while a memory arbiter will decide the running sequence according to the time when these requests are received by the processor.

According to our current version of SDK provided, SRAM space that is available to users is assigned as below:

- **RAM size < 36kB**, that is to say, when ESP8266EX is working under the station mode and is connected to the router, programmable space accessible to user in heap and data section is around 36kB.)
- There is no programmable ROM in the SoC, therefore, user program must be stored in an external SPI flash.

### 2.6.2. External SPI Flash

An external SPI flash is used together with ESP8266EX to store user programs. Theoretically speaking, up to 16 Mbyte memory capacity can be supported.

#### Suggested SPI Flash memory capacity:

- OTA is disabled: the minimum flash memory that can be supported is 512 kByte;
- OTA is enabled: the minimum flash memory that can be supported is 1 Mbyte.

Several SPI modes can be supported, including Standard SPI, Dual SPI, DIO SPI, QIO SPI, and Quad SPI.



Therefore, please choose the correct SPI mode when you are downloading into the flash, otherwise firmwares/programs that you downloaded may not work in the right way.

## 2.7. AHB and APB Blocks

The AHB blocks performs the function of an arbiter, controls the AHB interfaces from the MAC, SDIO (host) and CPU. Depending on the address, the AHB data requests can go into one of the two slaves: APB block, or flash controller (usually for standalone applications).

Data requests to the memory controller are usually high speed requests, and requests to the APB block are usually register access.

The APB block acts as a decoder. It is meant only for access to programmable registers within ESP8266's main blocks. Depending on the address, the APB request can go to the radio, SI/SPI, SDIO (host), GPIO, UART, real-time clock (RTC), MAC or digital baseband.



### 3. Pins and Definitions

The chipset encapsulates variable analog and data transmission I/Os, descriptions and definitions of which are explained below in detail.

#### 3.1. GPIO

##### 3.1.1. General Purpose Input/Output Interface (GPIO)

There are up to 17 GPIO pins. They can be assigned to various functions by the firmware. Each GPIO can be configured with internal pull-up (except XPD\_DCDC, which is configured with internal pull-down), input available for sampling by a software register, input triggering an edge or level CPU interrupt, input triggering a level wakeup interrupt, open-drain or push-pull output driver, or output source from a software register, or a sigma-delta PWM DAC.

These pins are multiplexed with other functions such as I2C, I2S, UART, PWM, IR Remote Control, etc. Data I/O soldering pad is bidirectional and tri-state that include data input and output controlling buffer. Besides, I/O can be set as a specific state and remains like this. For example, if you intend to lower the power consumption of the chip, all data input and output enable signals can be set as remaining low power state. You can transport some specific state into the I/O. When the I/O is not powered by external circuits, the I/O will remain to be the state that it was used the last time. Some positive feedback is generated by the state-remaining function of the pins, therefore, if the external driving power must be stronger than the positive feedback. Even so, the driving power that is needed is within 5uA.

Table 6 Pin Definitions of GPIOs

Variables	Symbol	Min	Max	Unit
Input Low Voltage	$V_{IL}$	-0.3	$0.25 \times V_{IO}$	V
Input High Voltage	$V_{IH}$	$0.75 \times V_{IO}$	3.3	V
Input Leakage Current	$I_{IL}$		50	nA
Output Low Voltage	$V_{OL}$		$0.1 \times V_{IO}$	V
Output High Voltage	$V_{OH}$	$0.8 \times V_{IO}$		V
Input Pin Resistance Value	$C_{pad}$		2	pF
VDDIO	$V_{IO}$	1.8	3.3	V
Maximum Driving Power	$I_{MAX}$		12	mA
Temerperature	$T_{amb}$	-40	125	°C

All digital IO pins are protected from over-voltage with a snap-back circuit connected between the pad and ground. The snap back voltage is typically about 6V, and the holding voltage is 5.8V. This



provides protection from over-voltages and ESD. The output devices are also protected from reversed voltages with diodes.

### 3.2. Secure Digital Input/Output Interface (SDIO)

One Slave SDIO has been defined by ESP8266EX, the definitions of which are described in Table 7 below. 4bit 25MHz SDIO v1.1 and 4bit 50MHz SDIO v2.0 are supported.

Table 7 Pin Definitions of SDIOs

Pin Name	Pin Num	IO	Function Name
SDIO_CLK	21	IO6	SDIO_CLK
SDIO_DATA0	22	IO7	SDIO_DATA0
SDIO_DATA1	23	IO8	SDIO_DATA1
SDIO_DATA_2	18	IO9	SDIO_DATA_2
SDIO_DATA_3	19	IO10	SDIO_DATA_3
SDIO_CMD	20	IO11	SDIO_CMD

### 3.3. Serial Peripheral Interface (SPI/HSPI)

Currently, one general Slave/Master SPI, one Slave SDID/SPI, and one general Slave/Master HSPI have been defined by ESP8266EX. Functions of all these pins can be implemented via hardware. The pin definitions are described below:

#### 3.3.1. General SPI (Master/Slave)

Table 8 Pin Definitions of General SPIs

Pin Name	Pin Num	IO	Function Name
SDIO_CLK	21	IO6	SPICLK
SDIO_DATA0	22	IO7	SPIQ/MISO
SDIO_DATA1	23	IO8	SPIID/MOSI
SDIO_DATA_2	18	IO9	SPIHD
SDIO_DATA_3	19	IO10	SPIWP
SDIO_CMD	20	IO11	SPICS0
U0TXD	26	IO1	SPICS1
GPIO0	15	IO0	SPICS2



### 3.3.2. SDIO / SPI (Slave)

Table 9 Pin Definitions of SDIO / SPI (Slave)

Pin Name	Pin Num	IO	Function Name
SDIO_CLK	21	IO6	SPI_SLAVE_CLK
SDIO_DATA0	22	IO7	SPI_SLAVE_MISO
SDIO_DATA1	23	IO8	SPI_SLAVE_INT
SDIO_DATA_2	18	IO9	NC
SDIO_DATA_3	19	IO10	SPI_SLAVE_CS
SDIO_CMD	20	IO11	SPI_SLAVE_MOSI

### 3.3.3. HSPI (Master/Slave)

Table 10 Pin Definitions of HSPI (Master/Slave)

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO14	HSPICLK
MTDI	10	IO12	HSPIQ/MISO
MTCK	12	IO13	HSPID/MOSI
MTDO	13	IO15	HPSICS

**Note:**

- SPI mode can be implemented via software programming. The clock frequency can reach up to a maximum value of 80MHz.
- Function of Slave SDIO/SPI interface can be implemented via hardware, and linked list DMA (Direct Memory Access) is supported, software overheads are smaller. However, there is no linked list DMA on general SPI and HSPI, and the software overheads are larger, therefore, the data transmitting speed will be restrained by software processing speed.

## 3.4. Inter-integrated Circuit Interface (I2C)

One I2C, which is mainly used to connect with micro controller and other peripheral equipment such as sensors, is defined by ESP8266EX. The present pin definition of I2C is as defined below:



Table 11 Pin Definitions of I2C

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO14	I2C_SCL
GPIO2	14	IO2	I2C_SDA

Both I2C-Master and I2C-Slave are supported. I2C interface functionality can be realized via software programming, the clock frequency can be up to around 100KHz at most. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

### 3.5. I2S

Currently one I2S data input interface and one I2S data output interface are defined. I2S interface is mainly used in applications such as data collection, processing, and transmission of audio data, as well as the input and output of serial data. For example, LED lights (WS2812 series) are supported. The pin definition of I2S is as defined below:

Table 12 Pin Definitions of I2S

I2S Data Input:			
Pin Name	Pin Num	IO	Function Name
MTDI	10	IO12	I2SI_DATA
MTCK	12	IO13	I2SI_BCK
MTMS	9	IO14	I2SI_WS

I2S Data Output:			
Pin Name	Pin Num	IO	Function Name
MTDO	13	IO15	I2SO_BCK
U0RXD	25	IO3	I2SO_DATA
GPIO2	14	IO2	I2SO_WS

I2S functionality can be realized via software programming, the GPIOs that will be used are multiplexed, and linked list DMA is supported.

### 3.6. Universal Asynchronous Receiver Transmitter (UART)

Two UART interfaces, UART0 and UART1, have been defined by ESP8266EX, the definitions are as below:



Table 13 Pin Definitions of UART Interfaces

Pin Type	Pin Name	Pin Num	IO	Function Name
UART0	U0RXD	25	IO3	U0RXD
	U0TXD	26	IO1	U0TXD
	MTDO	13	IO15	U0RTS
	MTCK	12	IO13	U0CTS
UART1	GPIO2	14	IO2	U1TXD
	SD_D1	23	IO8	U1RXD

Data transfers to/from UART interfaces can be implemented via hardware. The data transmission speed via UART interfaces can reach 115200\*40 (4.5Mbps).

UART0 can be for communication. It supports fluid control. Since UART1 features only data transmit signal (Tx), it is usually used for printing log.

Notes: By default, UART0 will output some printed information when the device is powered on and is booting up. The baud rate of the printed information is closely related to the frequency of the external crystal oscillator. If the frequency of the crystal oscillator is 40MHz, then the baud rate for printing is 115200; if the frequency of the crystal oscillator is 26MHz, then the baud rate for printing is 74880. If the printed information exerts any influence on the functionality of your device, you'd better block the printing during the power-on period by changing ([U0TXD](#), [U0RXD](#)) to ([MTDO](#), [MTCK](#)).

### 3.7. Pulse-Width Modulation (PWM)

Four PWM output interfaces have been defined by ESP8266EX. They can be extended by users themselves. The present pin definitions of the PWM interfaces are defined as below:

Table 14 Pin Definitions of PWM Interfaces

Pin Name	Pin Num	IO	Function Name
MTDI	10	IO12	PWM0
MTDO	13	IO15	PWM1
MTMS	9	IO14	PWM2
GPIO4	16	IO4	PWM3

The functionality of PWM interfaces can be implemented via software programming. For example, in the LED smart light demo, the function of PWM is realized by interruption of the timer, the minimum resolution can reach as much as 44 ns. PWM frequency range is adjustable from 1000 us to 10000 us,



i.e., between 100Hz and 1KHz. When the PWM frequency is at 1 KHz, the duty ratio will reach 1/22727, and over 14 bit resolution will be achieved at 1KHz refresh rate.

### 3.8. IR Remote Control

Currently, only one Infrared remote control interface is defined, the pin definition is as below:

Table 14 Pin Definition of IR Remote Control

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO12	IR Tx
GPIO5	24	IO5	IR Rx

The functionality of Infrared remote control interface can be implemented via software programming. NEC coding, modulation, and demodulation are used by this interface. The frequency of modulated carrier signal is 38KHz, while the duty ratio of the square wave is 1/3. The length of data transmission, which is around 1m, is determined by two factors: one is the maximum value of rated current, the other is internal current-limiting resistance value in the infrared receiver. The larger the resistance value, the lower the current, so is the power, and vice versa. The transmission angle is between 15° and 30°, and is mainly determined by the radiation direction of the infrared receiver.

**Notes:** Among the eight interfaces mentioned above, most of them can be multiplexed. Pin definitions that can be defined is not limited to the eight ones herein mentioned, customers can self customise the functions of the pins according to their specific application scenarios. Functions of these pins can be implemented via software programming and hardware.

### 3.9. ADC (Analog-to-digital Converter)

ESP8266EX is embedded with a 10-bit precision SARADC. Currently, TOUT (Pin6) is defined as ADC interface, the definition of which is described below:

Pin Name	Pin Num	Function Name
TOUT	6	ADC Interface

Table 16 Pin Definition of ADC

The following two applications can be implemented using ADC (Pin6). However, these two applications cannot be implemented concurrently.

- Test the power supply voltage of VDD3P3 (Pin 3 and Pin 4).

The function used to test the power supply voltage on PA\_VDD pin is: `uint16 system_get_vdd33(void)`

- Test the input voltage of TOUT (Pin 6):



The function used to test the input voltage of TOUT is: `uint16 system_adc_read(void)`

**RF-init** parameter in the following passage refers to `esp_init_data_default.bin`

**Application One:** Test the power supply voltage of VDD3P3 (Pin 3 and Pin 4).

**Hardware Design:** TOUT must be dangled.

**RF-init Parameter:** The 107th byte of `esp_init_data_default.bin` (0 - 127 byte), "vdd33\_const", must set to be 0xFF, i.e., the value of "vdd33\_const" is 255.

**RF Calibration Process:** Optimize the RF circuit conditions based on the testing results of VDD3P3 (Pin 3 and Pin 4).

**User Programming:** Use `system_get_vdd33` instead of `system_adc_read`.

**Application Two:** Test the input voltage of TOUT (Pin 6).

**Hardware Design:** The input voltage range is 0 to 1.0 V when TOUT is connected to external circuit.

**RF-init Parameter:** The value of the 107th byte of `esp_init_data_default.bin` (0 - 127 byte), "vdd33\_const", must be set to be the real power supply voltage of Pin 3 and Pin 4.

The working power voltage range of ESP8266EX is between 1.8V and 3.6V, while the unit of "vdd33\_const" is 0.1V, therefore, the effective value range of "vdd33\_const" is 18 to 36.

**RF Calibration Process:** Optimize the RF circuit conditions based on the value of "vdd33\_const". The permissible error is  $\pm 0.2V$ .

**User Programming:** Use `system_adc_read` instead of `system_get_vdd33`.

### Note One:

In `RF_init` parameter `esp_init_data_default.bin` (0 - 127 byte), the 107th byte is defined as "vdd33\_const". Definitions of "vdd33\_const" is described below:

(1) If `vdd33_const` = 0xff, the power voltage of Pin 3 and Pin 4 will be tested by the internal self-calibration process of ESP8266EX chipset itself. RF circuit conditions should be optimized according to the testing results.



(2) If  $18 \leq vdd33\_const \leq 36$ , ESP8266EX RF Calibration and optimization process is implemented via  $(vdd33\_const/10)$ .

(3) If  $vdd33\_const < 18$  or  $36 < vdd33\_const < 255$ , ESP8266EX RF Calibration and optimization process is implemented via the default value 3.0V.

#### Note Two:

Function `system_get_vdd33` is used to test the power supply voltage of VDD3P3 (Pin 3 and Pin 4). Details on this function are described below:

(1) Pin Tout must be dangled. The 107th byte of `esp_init_data_default.bin` (0 - 127 byte), "vdd33\_const", must set to be 0xFF.

(2) If the 107th byte of `esp_init_data_default.bin` (0 - 127 byte), "vdd33\_const", is equal to `0xff`, the returned value of function `system_get_vdd33` will be an effective value, otherwise `0xffff` will be returned.

(3) The unit of the returned value is: 1/1024 V.

#### Note Three:

Function `system_adc_read` is defined to test the input voltage of Pin TOUT (Pin 6). Details on this function are described below:

(1) The value of the 107th byte of `esp_init_data_default.bin` (0 - 127 byte), "vdd33\_const", must be set to be the real power supply voltage of Pin 3 and Pin 4.

(2) If the 107th byte of `esp_init_data_default.bin` (0 - 127 byte), "vdd33\_const", is NOT equal to `0xff`, the returned value of `system_adc_read` will be an effective value of the input voltage of Pin TOUT, otherwise `0xffff` will be returned.

(3) The unit of the returned value is: 1/1024 V.

## 3.10. LED Light and Button

ESP8266EX features up to 17 GPIOs, all of which can be assigned to realise various functions of LED lights and buttons. Definitions of some GPIOs that are assigned with certain functions in our demo application design are shown below:

Table 17 Pin Definitions of LED and Button

Pin Name	Pin Num	IO	Function Name
MTCK	12	IO13	Button (Reset)
GPIO0	15	IO0	WiFi Light
MTDI	10	IO12	Link Light



Altogether three interfaces have been defined, one is for the button, and the other two is for LED light. Generally, **MTCK** is used to control the reset button, **GPIO0** is used as an signal to indicate the WiFi working state, **MTDI** is used as a signal light to indicate communication between the device and the server.

Note: Among the nine interfaces mentioned above, most of them can be multiplexed. Pin definitions that can be defined is not limited to the eight ones herein mentioned, customers can self customise the functions of the pins according to their specific application scenarios. Functions of these pins can be implemented via software programming and hardware.



## 4. Firmware & Software Development Kit

The application and firmware is executed in on-chip ROM and SRAM, which loads the instructions during wake-up, through the SDIO interface, from the external flash.

The firmware implements TCP/IP, the full 802.11 b/g/n/e/i WLAN MAC protocol and WiFi Direct specification. It supports not only basic service set (BSS) operations under the distributed control function (DCF) but also P2P group operation compliant with the latest WiFi P2P protocol. Low level protocol functions are handled automatically by ESP8266:

- RTS/CTS
- acknowledgement
- fragmentation and defragmentation
- aggregation
- frame encapsulation (802.11h/RFC 1042)
- automatic beacon monitoring / scanning, and
- P2P WiFi direct

Passive or active scanning, as well as P2P discovery procedure is performed autonomously once initiated by the appropriate command. Power management is handled with minimum host interaction to minimize active duty period.

### 4.1. Features

The SDK includes the following library functions:

- 802.11 b/g/n/d/e/i/k/r support;
- WiFi Direct (P2P) support;
- P2P Discovery, P2P Group Owner mode, P2P Power Management
- Infrastructure BSS Station mode / P2P mode / softAP mode support;
- Hardware accelerators for CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WAPI (SMS4), WEP (RC4), CRC;
- WPA/WPA2 PSK, and WPS driver;
- Additional 802.11i security features such as pre-authentication, and TSN;
- Open Interface for various upper layer authentication schemes over EAP such as TLS, PEAP, LEAP, SIM, AKA, or customer specific;
- 802.11n support (2.4GHz);
- Supports MIMO 1×1 and 2×1, STBC, A-MPDU and A-MSDU aggregation and 0.4µs guard interval;



- WMM power save U-APSD;
- Multiple queue management to fully utilize traffic prioritization defined by 802.11e standard;
- UMA compliant and certified;
- 802.1h/RFC1042 frame encapsulation;
- Scattered DMA for optimal CPU off load on Zero Copy data transfer operations;
- Antenna diversity and selection (software managed hardware);
- Clock/power gating combined with 802.11-compliant power management dynamically adapted to current connection condition providing minimal power consumption;
- Adaptive rate fallback algorithm sets the optimum transmission rate and Tx power based on actual SNR and packet loss information;
- Automatic retransmission and response on MAC to avoid packet discarding on slow host environment;
- Seamless roaming support;
- Configurable packet traffic arbitration (PTA) with dedicated slave processor based design provides flexible and exact timing Bluetooth co-existence support for a wide range of Bluetooth Chip vendors;
- Dual and single antenna Bluetooth co-existence support with optional simultaneous receive (WiFi/Bluetooth) capability.

## 5. Power Management

The chip can be put into the following states:

- **OFF:** CHIP\_PD pin is low. The RTC is disabled. All registers are cleared.
- **DEEP\_SLEEP:** Only RTC is powered on - the rest of the chip is powered off. Recovery memory of RTC can keep basic WiFi connecting information.
- **SLEEP:** Only the RTC is operating. The crystal oscillator is disabled. Any wakeup events (MAC, host, RTC timer, external interrupts) will put the chip into the WAKEUP state.
- **WAKEUP:** In this state, the system goes from the sleep states to the PWR state. The crystal oscillator and PLLs are enabled.
- **ON:** the high speed clock is operational and sent to each block enabled by the clock control register. Lower level clock gating is implemented at the block level, including the CPU, which can be gated off using the WAITI instruction, while the system is on.

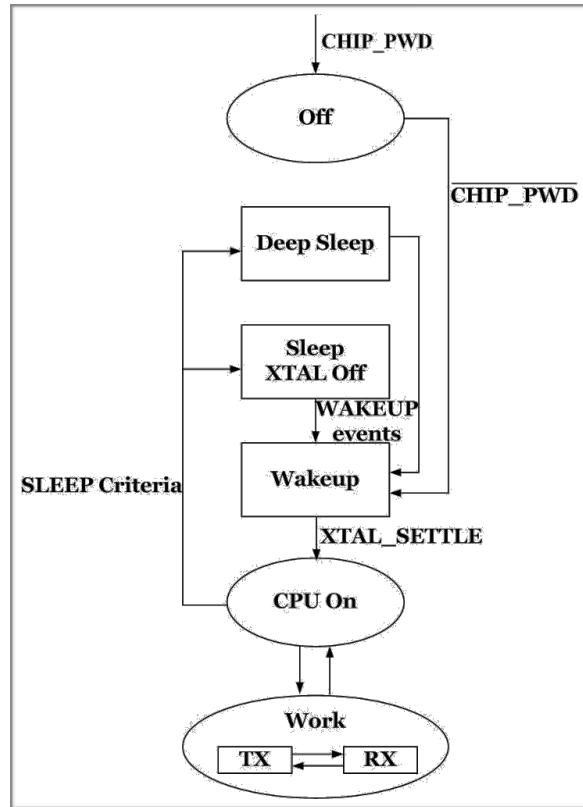


Figure 3 Illustration of Power Management

## 6. Clock Management

### 6.1. High Frequency Clock

The high frequency clock on ESP8266EX is used to drive both transmit and receive mixers. This clock is generated from the internal crystal oscillator and an external crystal. The crystal frequency can range from 26MHz to 52MHz.

While internal calibration of the crystal oscillator ensures that a wide range of crystals can be used, in general, the quality of the crystal is still a factor to consider, to have reasonable phase noise that is required for good performance. When the crystal selected is sub-optimal due to large frequency drifts or poor Q-factor, the maximum throughput and sensitivity of the WiFi system is degraded. Please refer to the application notes on how the frequency offset can be measured.



Table 18 High Frequency Clock

Parameter	Symbol	Min	Max	Unit
Frequency	FXO	26	52	MHz
Loading capacitance	CL		32	pF
Motional capacitance	CM	2	5	pF
Series resistance	RS	0	65	$\Omega$
Frequency tolerance	$\Delta$ FXO	-15	15	ppm
Frequency vs temperature (-25°C ~ 75°C)	$\Delta$ FXO,Temp	-15	15	ppm

## 6.2. External Reference Requirements

For an externally generated clock, the frequency can range from 26MHz to 52MHz can be used. For good performance of the radio, the following characteristics are expected of the clock:

Table 19 External Clock Reference

Parameter	Symbol	Min	Max	Unit
Clock amplitude	VXO	0.2	1	Vpp
External clock accuracy	$\Delta$ FXO,EXT	-15	15	ppm
Phase noise @1kHz offset, 40MHz clock			-120	dBc/Hz
Phase noise @10kHz offset, 40MHz clock			-130	dBc/Hz
Phase noise @100kHz offset, 40MHz clock			-138	dBc/Hz

## 7. Radio

The ESP8266EX radio consists of the following main blocks:

- 2.4GHz receiver
- 2.4GHz transmitter
- High speed clock generators and crystal oscillator
- Real time clock
- Bias and regulators
- Power management



## 7.1. Channel Frequencies

The RF transceiver supports the following channels according to the IEEE802.11b/g/n standards.

Table 20 Frequency Channel

Channel No	Frequency (MHz)	Channel No	Frequency (MHz)
1	2412	8	2447
2	2417	9	2452
3	2422	10	2457
4	2427	11	2462
5	2432	12	2467
6	2437	13	2472
7	2442	14	2484

## 7.2. 2.4 GHz Receiver

The 2.4GHz receiver downconverts the RF signal to quadrature baseband signals and converts them to the digital domain with 2 high resolution high speed ADCs. To adapt to varying signal channel conditions, RF filters, automatic gain control (AGC), DC offset cancelation circuits and baseband filters are integrated within ESP8266EX.

## 7.3. 2.4 GHz Transmitter

The 2.4GHz transmitter up-converts the quadrature baseband signals to 2.4GHz, and drives the antenna with a high powered CMOS power amplifier. The use of digital calibration further improves the linearity of the power amplifier, enabling a state of art performance of delivering +19.5dBm average power for 802.11b transmission and +16dBm for 802.11n transmission.

Additional calibrations are integrated to cancel any imperfections of the radio, such as:

- carrier leakage,
- I/Q phase matching, and
- baseband nonlinearities

This reduces the amount of time required and test equipment required for production testing.

## 7.4. Clock Generator

The clock generator generates quadrature 2.4 GHz clock signals for the receiver and transmitter. All components of the clock generator are integrated on-chip, including:



- inductor,
- varactor, and
- loop filter

The clock generator has built-in calibration and self test circuits. Quadrature clock phases and phase noise are optimized on-chip with patented calibration algorithms to ensure the best receiver and transmitter performance.

## 8. Appendix: QFN32 Package Size

