OPB780-Kit

Color Sensor Evaluation Kit

- **The OPB780-Kit is designed to provide the design engineer an easy way to evaluate the capability of the OPB780 Color Sensor.**

The **OPB780Z** is a full color sensor with 4 different frequencies relating directly to a specific color seen by the sensor.

The **OPB780Z** color sensor uses a programmable light-to-frequency converter that combines 64 configurable silicon photodiodes (on a 144 um center array and measuring 120 um x 120 um each) and a current-to-frequency converter on a single monolithic CMOS integrated circuit, packaged in a small, lightweight package that makes it ideal for using in miniature applications.

The output is a square wave (50% duty cycle) with a frequency directly proportional to light chromaticity and irradiance from four filtered diode arrays.

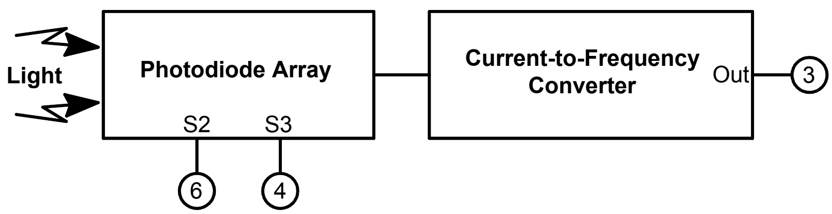
The light-to-frequency converter reads an 8 x 8 array of photodiodes that consists of four groups of 16 photodiodes each, segregated by color: 16 photodiodes with red filters, 16 photodiodes with green filters, 16 photodiodes with blue filters and 16 photodiodes with clear filters. Each color's group of 16 photodiodes is interdigitated to minimize the effect of non-uniformity of incident irradiance. Each color's group is also connected in parallel. The type of photodiode used during operation is pin-selectable.

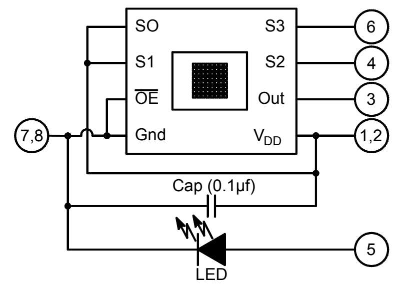
The internal photodiode used by the device is controlled by two logic inputs, S2 and S3.

Pin 4 (S2)	Pin 6 (S3)	Diode Filter
H	H	Green
H	L	Clear
L	L	Red
L	H	Blue

The output of the device is designed to drive a standard TTL or CMOS logic input over short distances.

Block





Pin Name	Pin #	Description
V_{DD}	1, 2	Supply voltage
OUT	3	Output Frequency (F_O)
S2	4	Photodiode type selection input
LED Anode	5	LED input
S3	6	Photodiode type selection input
GND/OE	7, 8	Sensor & LED Ground

DO NOT LOOK DIRECTLY AT LED WITH UNSHIELDED EYES OR DAMAGE TO RETINA MAY OCCUR.

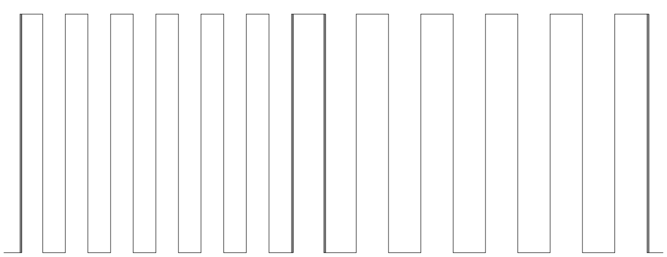
The **OPB780Z** includes 10" [25.4cm] a Flat Flexible interface Cable. Custom lengths of the cable can be ordered either from OPTEK or most Flat Flexible Cable (FFC) manufactures. The FFC is designed to interface with an any standard 0.5mm spacing FFC connector similar to the AVX ELCO connector.
(part number 04 6249 0080 00 800+).

The **OPB780Z** consist of:

1. White LED for illumination of the target
2. Photodiode Array consisting of:
   - 16 Green Filter Diodes
   - 16 Clear Filter Diodes
   - 16 Red Filter Diodes
   - 16 Blue Filter Diodes
3. Current to Frequency Converter
4. TTL and CMOS Drive Circuitry
5. Plastic Package with 10: (25.4cm) FFC

All 16 photodiodes that have the same color filter are connected in parallel. Package pin numbers 4 (S2) and 6 (S3) can be used to select the desired photodiode/filter configuration to optimize color identification. All 64 photodiodes are 120µm x 120µm in size and are on 144µm centers.

The output frequency provided by the **OPB780Z** may vary from part to part and should be calibrated as required to meet each application. The more colors that are to be identified require more accurate calibration and longer sampling rate. A small amount of jitter will be present in the frequency provided for a color and must be taken into consideration depending on the accuracy required.



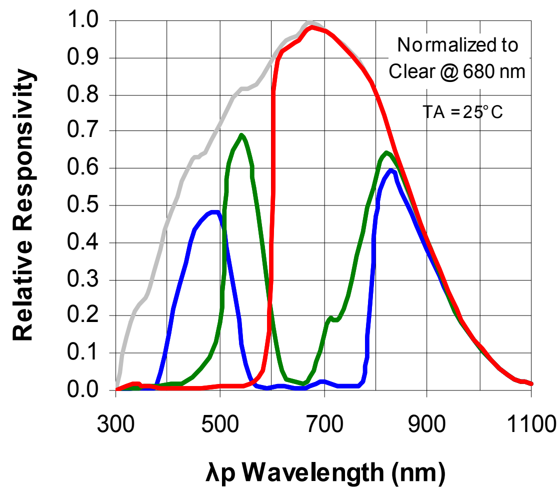
Frequency-A Frequency-B

The internal scaling is preconfigured for the highest frequency output of typically 600kHz. The 600kHz frequency shortens the sampling time providing a small amount of jitter in the output frequency.

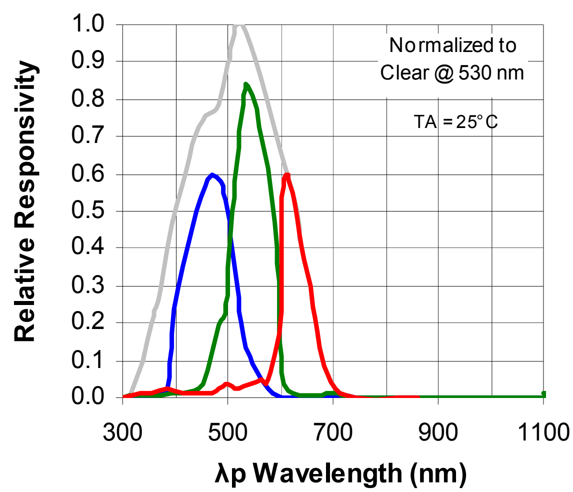
The user must take into consideration the luminance intensity (amount of light) that may be acknowledged by the human eye. As the luminance intensity decreases, the amount of color recognized is diminished. In order to consistently reproduce a specific color with the same output frequency, the luminance intensity must be the consistent. When critical identification of specific colors is required, the angle of view and distance may critical characteristics.

The **OPB780Z** photodiodes as provided are designed to recognize ultraviolet, visible as well as near infrared wavelengths (from 300 to 1,050 nanometers) The graphs below show the typical Spectral Response for each photodiode configuration. When a near infrared rejecting filter similar to the Hoya CM500 is placed in front of the sensor, the higher wavelengths (above 700 nm) are eliminated thus enhancing recognition of colors in the visible range.

Typical Spectral Response
Full Wavelength Sensitivity



Spectral Response



The standard product allows the **OPB780Z** to be used to recognize colors over the full wavelength of 300nm to 1100nm beyond those of the human visual range.

OPTEK Technology Inc. — 1645 Wallace Drive, Carrollton, Texas 75006
Phone: (972) 323-2200 or (800) 341-4747 FAX: (972) 323-2396 sensors@optekinc.com www.optekinc.com

Below are the operating characteristics of the **OPB780Z**. As the ambient temperature or other environmental conditions change, the values noted may vary. The **OPB780Z** is designed to operate between –40°C and +85°C. The **OPB780Z** provides consistent recognition of a color given constant environmental conditions.

Operating Characteristics at V_{DD} = 5 V, T_A = 25°C

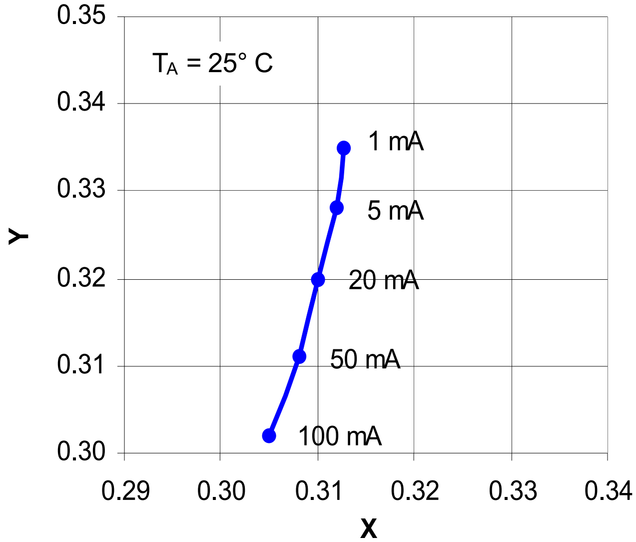
Parameter	Test Conditions	Clear Diode			Blue Diode			Green Diode			Red Diode			Units
		S2 = H, S3 = L			S2 = L, S3 = H			S2 = H, S3 = H			S2 = L, S3 = L			
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Output Frequency (f_O)	E_e = 47.2 µW/cm^2 λ_P = 470 nm	16.0	20.0	24.0	11.2	16.4	21.6	-	-	-	-	-	-	kHz
	E_e = 40.4 µW/cm^2 λ_P = 524 nm	16.0	20.0	24.0	-	-	-	8.0	13.6	19.2	-	-	-	kHz
	E_e = 34.6 µW/cm^2 λ_P = 640	16.0	20.0	24.0	-	-	-	-	-	-	14.0	19.0	24.0	kHz
Dark Frequency	E_e = 0	-	2	12	-	2	12	-	2	12	-	2	12	Hz
Irradiance Responsivity (R_e) (See Note 1)	λ_P = 470 nm	-	424	-	-	348	-	-	81	-	-	26	-	Hz / (µW/cm^2)
	λ_P = 524 nm	-	495	-	-	163	-	-	337	-	-	35	-	
	λ_P = 565 nm	-	532	-	-	37	-	-	309	-	-	91	-	
	λ_P = 640 nm	-	578	-	-	17	-	-	29	-	-	550	-	
Saturation Irradiance (See Note 2)	λ_P = 470 nm	-	1410	-	-	1720	-	-	-	-	-	-	-	(µW/cm^2)
	λ_P = 524 nm	-	1210	-	-	-	-	-	1780	-	-	-	-	
	λ_P = 565 nm	-	1130	-	-	-	-	-	1940	-	-	-	-	
	λ_P = 640 nm	-	1040	-	-	-	-	-	-	-	-	1090	-	
Illuminance Responsivity (R_V) (See Note 3)	λ_P = 470 nm	-	565	-	-	464	-	-	108	-	-	35	-	Hz / Lux
	λ_P = 524 nm	-	95	-	-	31	-	-	65	-	-	7	-	
	λ_P = 565 nm	-	89	-	-	6	-	-	52	-	-	15	-	
	λ_P = 640 nm	-	373	-	-	11	-	-	19	-	-	355	-	
Nonlinearity (See Note 4)	(f_O) = 0 to 5 kHz	-	±0.1	-	-	±0.1	-	-	±0.1	-	-	±0.1	-	% Full Scale
	(f_O) = 0 to 50 kHz	-	±0.2	-	-	±0.2	-	-	±0.2	-	-	±0.2	-	
	(f_O) = 0 to 500 kHz	-	±0.5	-	-	±0.5	-	-	±0.5	-	-	±0.5	-	

Notes:
1. Irradiance Responsivity (R_e) is measured over the range from Frequency 0 to 5 kHz.
2. Saturation Irradiance—(Full-Scale Frequency) / (Irradiance Responsivity)
3. Illuminance Responsivity is calculated from the Irradiance Responsivity using the appropriate wavelength LED.
4. Nonlinearity is defined as the deviation of the Output Frequency utilizing a straight line between 0 and full scale and is expressed as a percent of full scale.

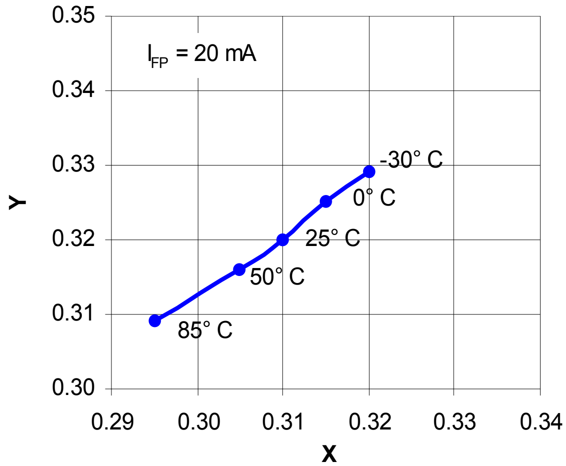
One of the critical parts of the **OPB780Z** is the **LED**. It is very important to understand its characteristics relative to forward current and temperature. The below chart shows how the forward current of the LED changes the Chromatic coordinates.

As can be seen, increasing the forward current of the LED lowers the "Y" value while minimally changing the "X" value. The projected most optimum light (equivalent to sun light at 12 noon with no contaminates in the air) has both "X" and "Y" chromatic values of 1/3. Chromatic numbers below 1/3 consist of more Blue or Green wavelengths and numbers above 1/3 consist of more Red or Yellow wavelengths. As can be seen a forward current of approximately 4 mA would provide the most optimum "Y" value.

Forward Current vs Chromaticity Coordinate



Ambient Temperature vs Chromaticity Coordinate

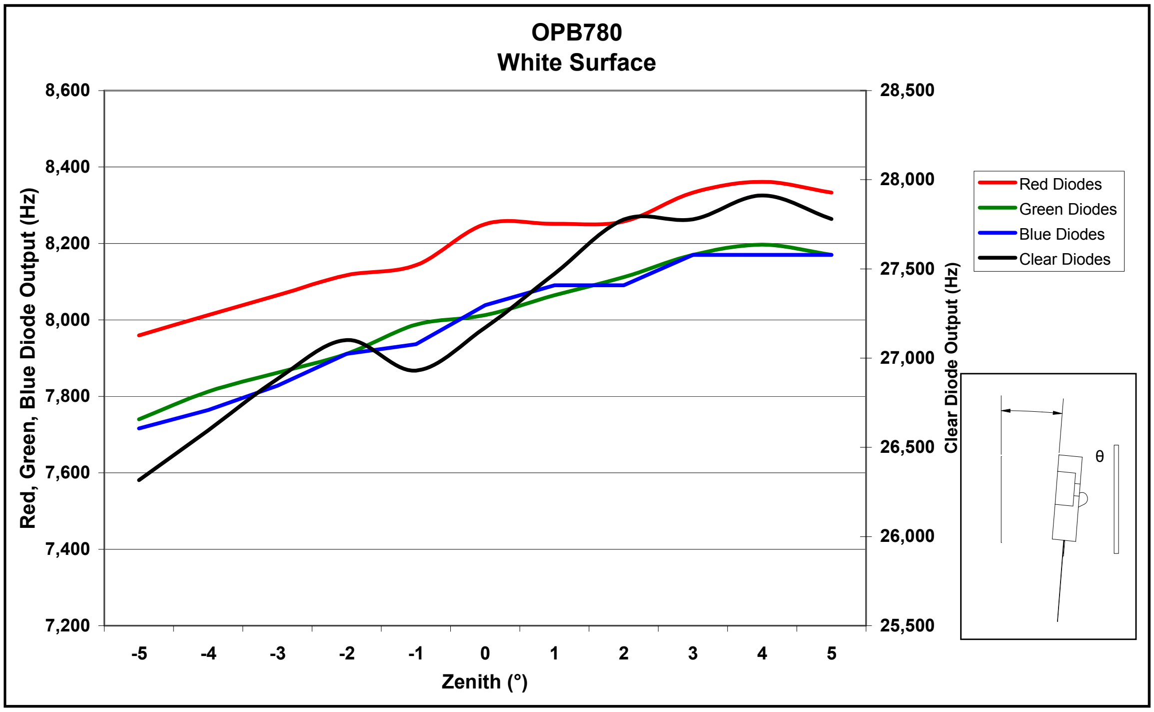


The graphs show the change in chromaticity vs temperature. As the temperature reduces the "X" value moves closer toward the optimum (1/3) value.
In conclusion, the lower the forward current and the cooler the temperature, the closer the LED transmits to the optimum chromatic values of "X"=1/3 and "Y"=1/3.

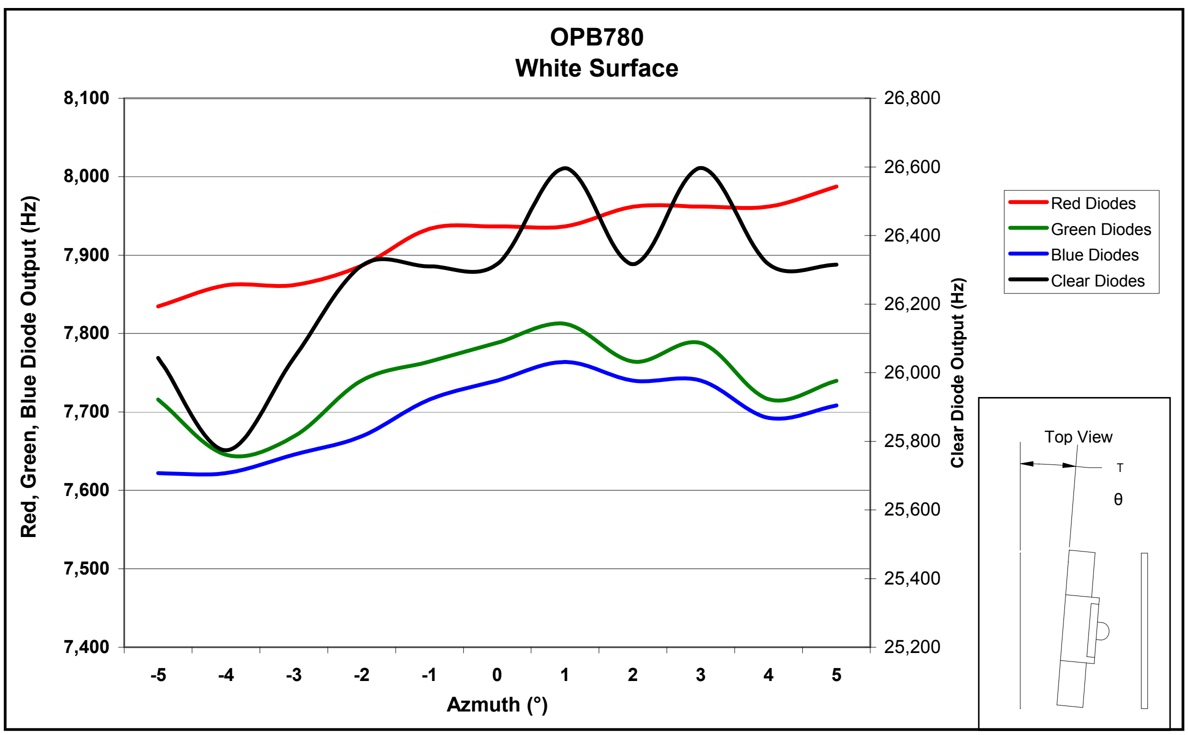
In order to provide sufficient light levels and the best color resolution, the white LED should be driven with at least 20 mA of forward current and operated at room temperatures (approx. 25°C). These conditions result in chromatic ranges of about "X"=0.31 and "Y"=0.32.

The responsivity of the each diode group is dependent on the angular position of the device to the object. The graphs showing the typical variance of the Output frequency of the device vs both Azmuth (twist) and Zenith (bend) are shown in the two graphs.

OPTEK Technology Inc. — 1645 Wallace Drive, Carrollton, Texas 75006
Phone: (972) 323-2200 or (800) 341-4747 FAX: (972) 323-2396 sensors@optekinc.com www.optekinc.com



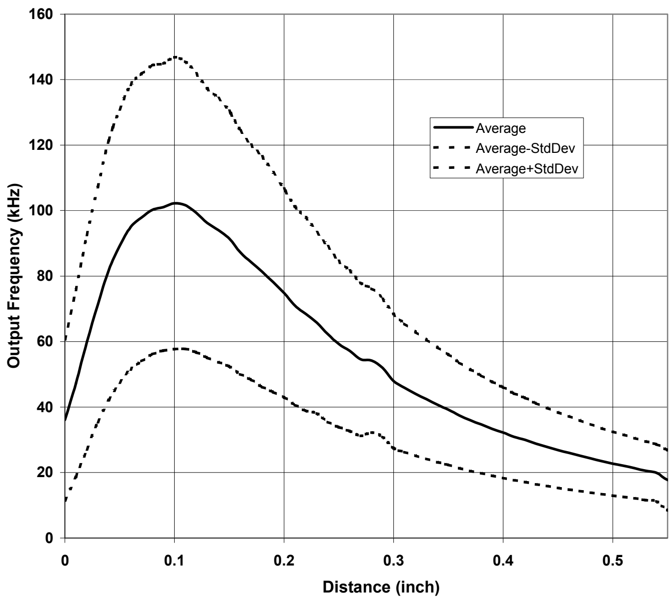
OPB780
White Surface
Red, Green, Blue Diode Output (Hz)
8,600
8,400
8,200
8,000
7,800
7,600
7,400
7,200
Clear Diode Output (Hz)
28,500
28,000
27,500
27,000
26,500
26,000
25,500
-5 -4 -3 -2 -1 0 1 2 3 4 5
Zenith (°)
Red Diodes
Green Diodes
Blue Diodes
Clear Diodes
θ



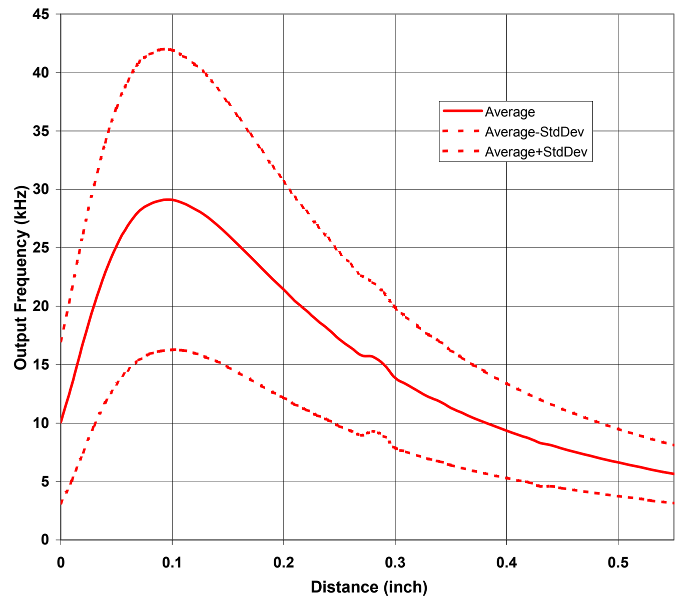
OPB780
White Surface
Red, Green, Blue Diode Output (Hz)
8,100
8,000
7,900
7,800
7,700
7,600
7,500
7,400
Clear Diode Output (Hz)
26,800
26,600
26,400
26,200
26,000
25,800
25,600
25,400
25,200
-5 -4 -3 -2 -1 0 1 2 3 4 5
Azmuth (°)
Red Diodes
Green Diodes
Blue Diodes
Clear Diodes
Top View
θ

The distance of the object from the sensor is a critical parameter with typical frequency responses for each Diode and filter combination graphed below.

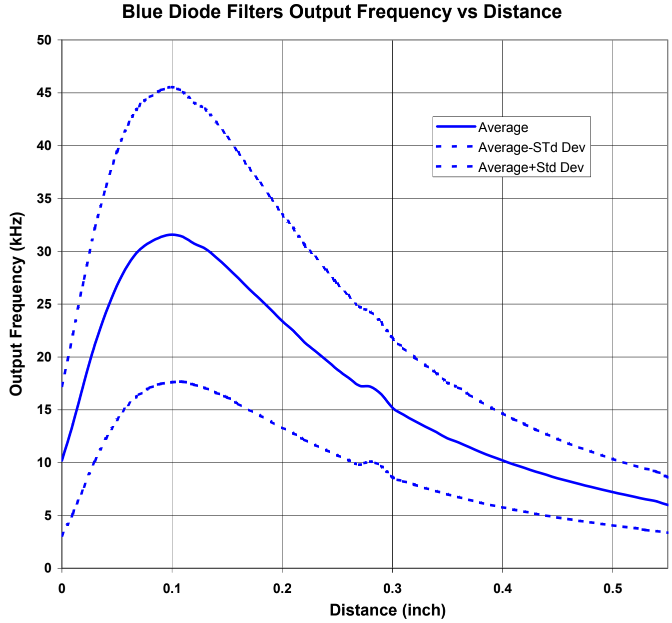
Clear Diode Filter Output Frequency vs Distance



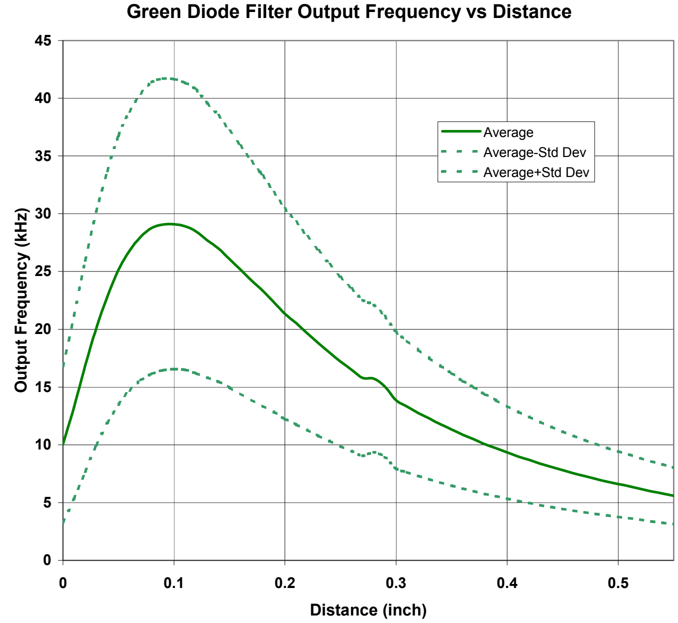
Red Diode Filters Output Frequency vs Distance



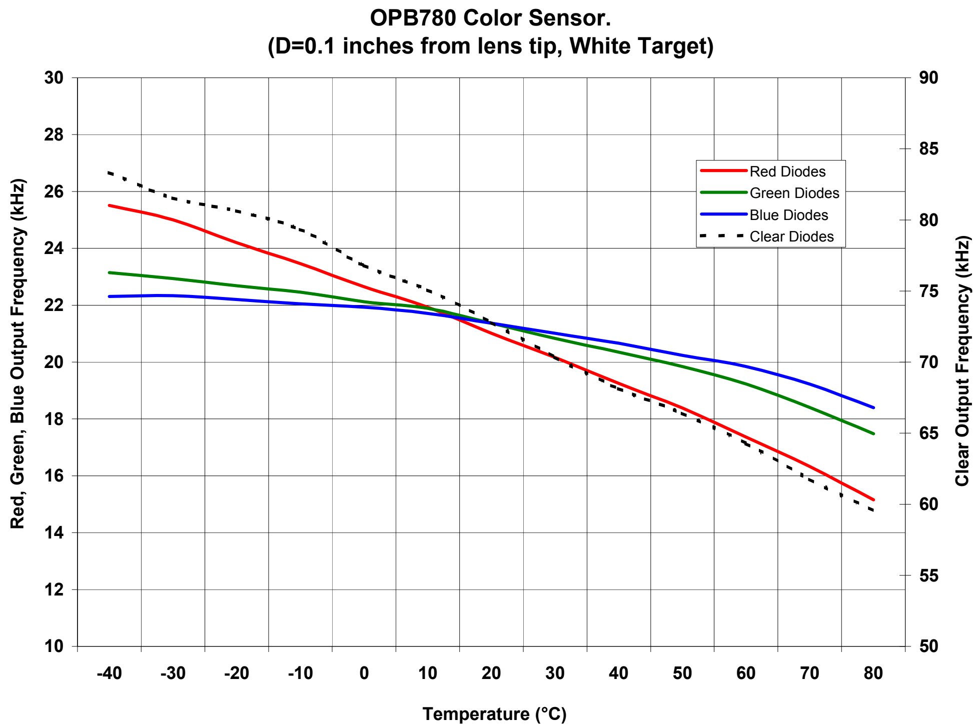
Blue Diode Filters Output Frequency vs Distance

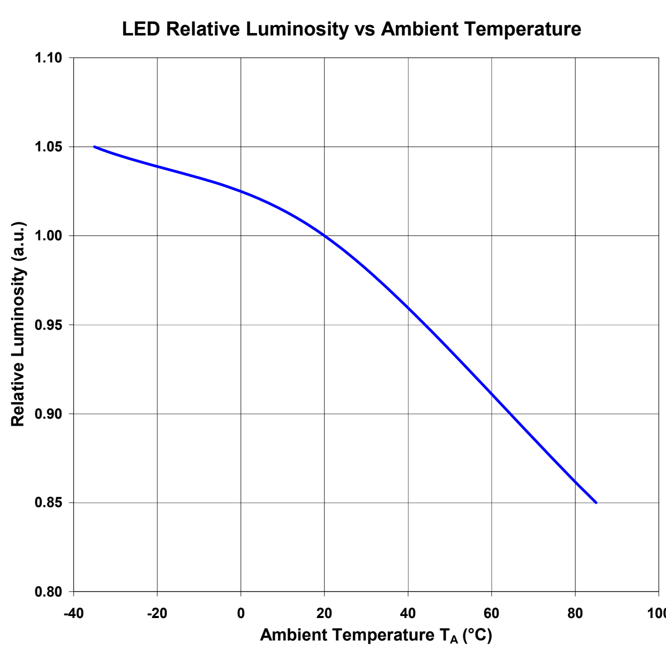


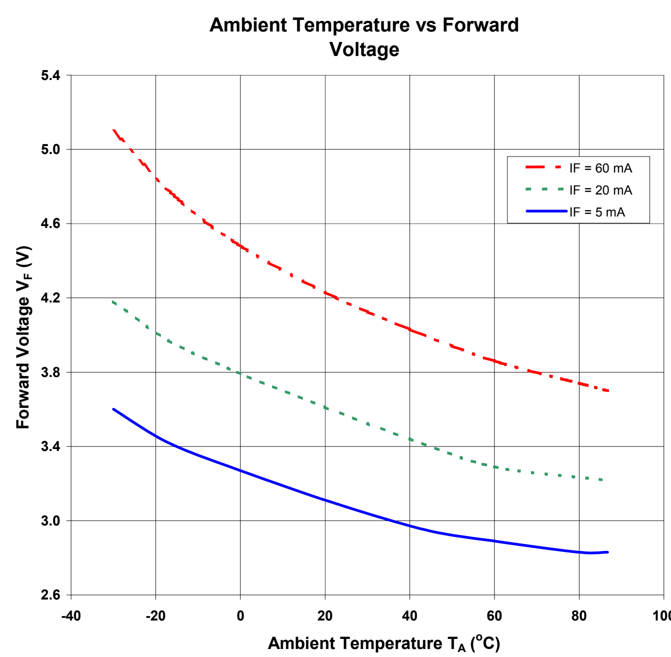
Green Diode Filter Output Frequency vs Distance

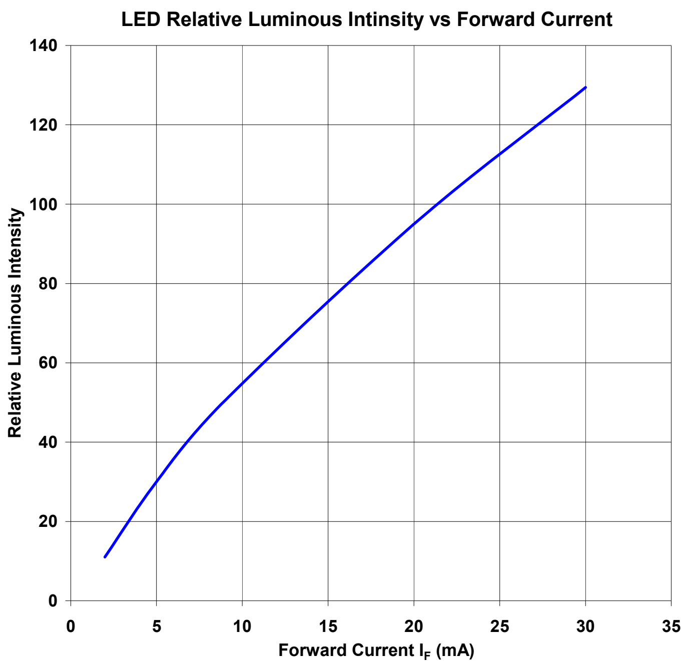


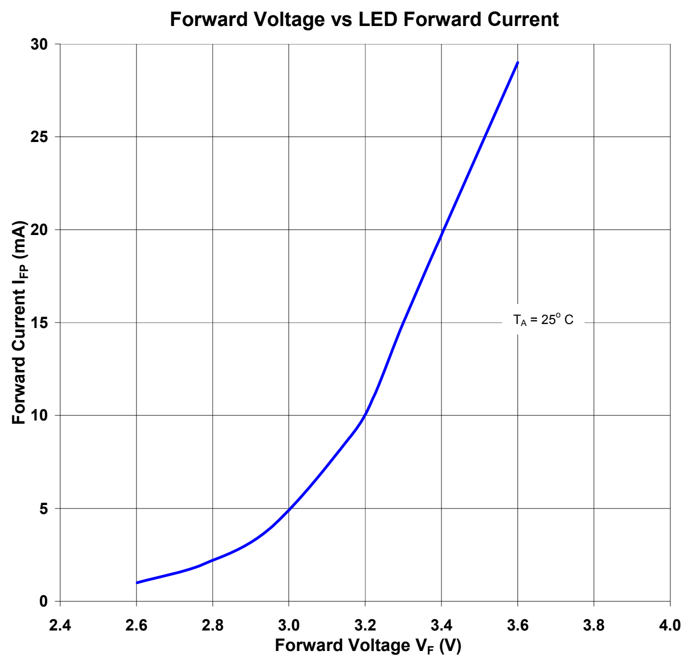
OPTEK Technology Inc. — 1645 Wallace Drive, Carrollton, Texas 75006
Phone: (972) 323-2200 or (800) 341-4747 FAX: (972) 323-2396 sensors@optekinc.com www.optekinc.com

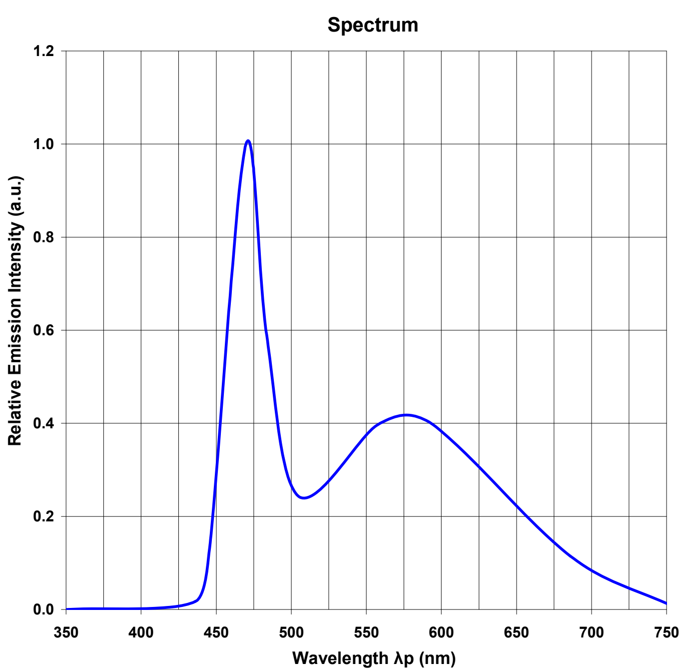

OPB780 Color Sensor.
(D=0.1 inches from lens tip, White Target)
Red, Green, Blue Output Frequency (kHz)
Clear Output Frequency (kHz)
Temperature (°C)
Red Diodes
Green Diodes
Blue Diodes
Clear Diodes
30
28
26
24
22
20
18
16
14
12
10
90
85
80
75
70
65
60
55
50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80

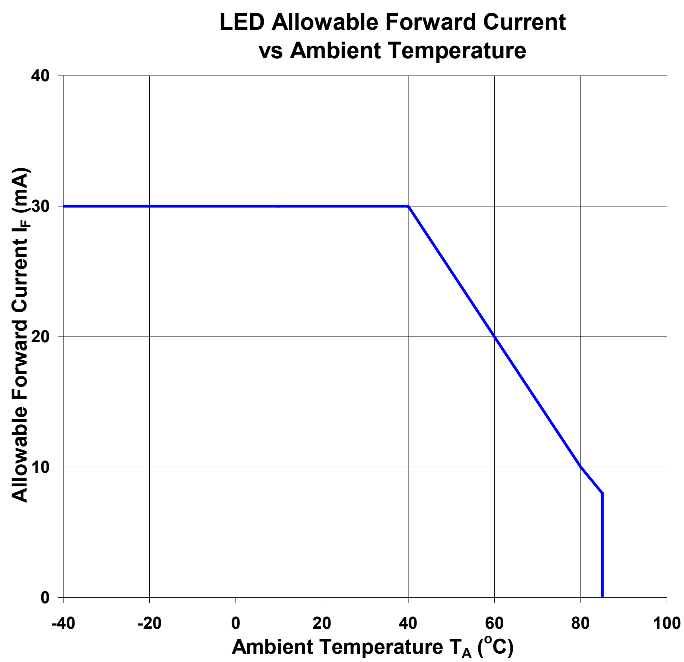

LED Relative Luminosity vs Ambient Temperature
Relative Luminosity (a.u.)
Ambient Temperature TA (°C)
1.10
1.05
1.00
0.95
0.90
0.85
0.80
-40
-20
0
20
40
60
80
100


Ambient Temperature vs Forward Voltage
Forward Voltage VF (V)
Ambient Temperature TA (°C)
IF = 60 mA
IF = 20 mA
IF = 5 mA
5.4
5.0
4.6
4.2
3.8
3.4
3.0
2.6
-40
-20
0
20
40
60
80
100



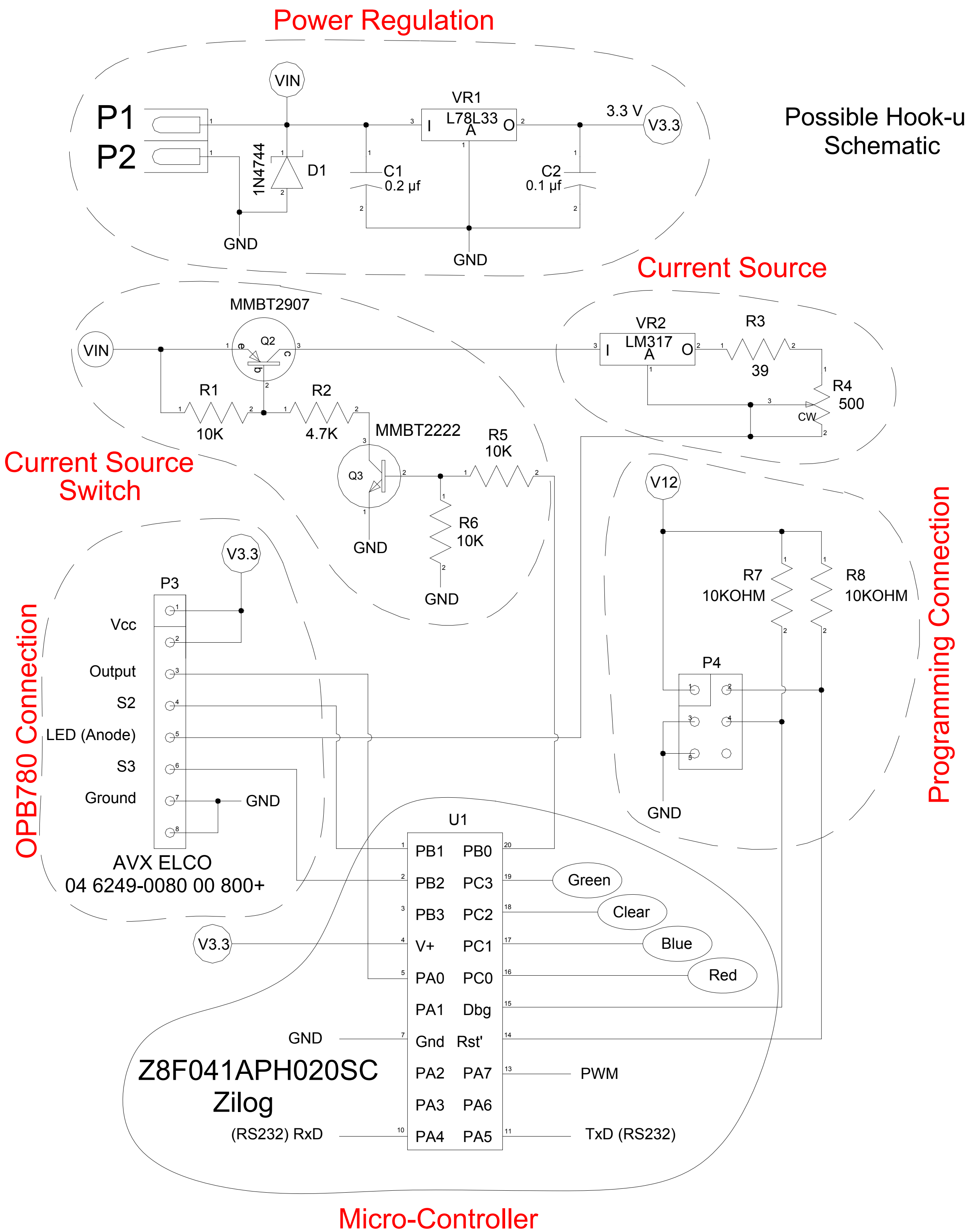






OPTEK reserves the right to make changes at any time in order to improve design and to supply the best product possible.

OPTEK Technology Inc. — 1645 Wallace Drive, Carrollton, Texas 75006
Phone: (972) 323-2200 or (800) 341-4747 FAX: (972) 323-2396 sensors@optekinc.com www.optekinc.com



OPTEK reserves the right to make changes at any time in order to improve design and to supply the best product possible.

Main Programming C-code for the Circuit on page 9 .

```
/***********************************************************************
*  Copyright (C) 2006 by OPTEK Technology
*  All Rights Reserved
*  This code is provided as a tool for design engineers and OPTEK assumes no responsibility or liability.
**********************************************************************/

/**********************************************************************
* This program is for the Color Sensor display
**********************************************************************/

#include <eZ8.h>
#include <defines.h>
#include "proto.h"
#include "refer.h"


/*********************************************************
 Main program begins here
*********************************************************/

void main (void)
{
	unsigned char u8LoopCount;	// software timing loop counter
	unsigned char u16Clear;	// Count of rising edges seen on clear sensor
	unsigned char u16Blue;	// Count of rising edges seen on blue sensor
	unsigned char u16Green;	// Count of rising edges seen on Green sensor
	unsigned char u16Red;	// Count of rising edges seen on Red sensor
	unsigned char u8ColorSelect;	// This is used to keep track of sensor being monitored

	init_gpio();	// Initializes LED ports (Port A and C)
	init_timer0();	// Initialize timer/counter 0
	DI();	// Disable Interrupts
	EI();	// Enable Interrupts

	u8LoopCount = 0;
	u16Red = 0;
	u16Blue = 0;
	u16Clear = 0;
	u16Green = 0;
	u8ColorSelect = 0;

	PortBBitsOn (0x01);	// turn on White LED
	Delmsec(10);	// Startup delay

	while(1)	// Infinite while loop
	{
		u8LoopCount ++;
		if (u8LoopCount == 0)	// Check for roll-over
		{

			//Read Sensor Output frequency count into correct location
			ReadTimer0 ();

			switch(u8ColorSelect)
```


OPTEK Technology Inc. — 1645 Wallace Drive, Carrollton, Texas 75006
Phone: (972) 323-2200 or (800) 341-4747 FAX: (972) 323-2396 sensors@optekinc.com www.optekinc.com

```
				{
					case 0:
						u16Red = uval.WSized.tmw;
						break;

					case 1:
						u16Blue = uval.WSized.tmw;
						break;

					case 2:
						u16Clear = uval.WSized.tmw;
						break;

					case 3:
						u16Green = uval.WSized.tmw;
						break;

					default:
						break;
				}

				PortBBitsOff (u8ColorSelect * 2);	//set S2 & S3 to 0
				u8ColorSelect ++;			//Advance color counter
				if (u8ColorSelect > 3)		//and check for roll-over
					u8ColorSelect = 0;

				PortBBitsOn (u8ColorSelect * 2);	//set up S2 & S3 for next color

				SetTimer0();				//reset timer0

				PCOUT = 0xff;				//turn on all LED's
			}

			if (u8LoopCount >= u16Red)		//check for Red turn off
				PCOUT &= 0xFE;

			if (u8LoopCount >= u16Blue)		//check for Blue turn off
				PCOUT &= 0xFD;

			if (u8LoopCount >= u16Clear)		//check for Clear turn off
				PCOUT &= 0xFB;

			if (u8LoopCount >= u16Green)		//check for Green turn off
				PCOUT &= 0xF7;
		}
}


/***************************************************************
PortBBitsOn turns on 1 or more bits without disturbing other bits
***************************************************************/
void PortBBitsOn (unsigned char u8BitNums)
{
	PBOUT |= u8BitNums;
}
```

```
/****************************************************************
PortBBitsOff turns off 1 or more bits without disturbing other bits
*****************************************************************/
void PortBBitsOff (unsigned char u8BitNums)
{
        PBOUT &= (0xff ^ u8BitNums);
}
```

General Purpose I/O C-code referenced by Main Program

```
/**************************************************
*  Copyright (C) 2006 by OPTEK Technology
 *  All Rights Reserved
*  This code is provided as a tool for design engineers and OPTEK assumes no responsibility or liability.
 **************************************************/

#include <eZ8.h>

///////////////////////////////////////////////////////
// Initializes ports
//

void init_gpio(void)
{
        /**************************************************/
        //Port A Data Direction Register(01): Reset value = 0xff
        //0=Output, 1=Input
        PAADDR        = 0x01;    // Select PortA Data Dir Register
        PACTL = 0xff;  // 1 1 0 1    1 1 1 1

        //Port A Alternate Function Select (07): Reset value=00h
        //Note: Since only 1 function per pin, AFS1 and AAFS2 registers are not used:
        //                just enable Alternate Function
                          //PA0:   T0In/T0OutNot
                          //PA1:  T0Out
                          //PA2:   DE0
                          //PA3:   CtsNot
                          //PA4:   Rxd0/IrRx0
                          //PA5:   Txd0/IrTx0
                          //PA6:   T1In/T1OutNot
                          //PA7:   T1Out
        PAADDR = 0x07;          // Select AFS1 register
        PACTL  = 0x01;          // and write data to it

        PAADDR = 0x08;          // Select AFS2 register
        PACTL  = 0x00;          // data to write to AFS2 register

        //Port A Output Control (03): Reset value = 0x00
        //0 = Enable Active pullup for all output modes (Active Drain output)
        //1 = Disable Active pullup (Open Drain output)
        PAADDR = 0x03;
```

```
PACTL = 0xff;

//Port A High Drive Enable (0x04): Reset value = 0x00
//0 = High output drive disabled
//1 = High output drive enabled
PAADDR = 0x04;
PACTL = 0x00;

//Port A STOPMODE recovery (0x05): Reset value = 0x??
//0 = Stop Mode Recovery is disabled for this bit
//1 = Stop Mode Recovery is enabled for this bit
PAADDR = 0x05;
PACTL = 0x00;

//Port A Pullup Enable (0x06): Reset value = 0x00
//0 = Weak Pullup is disabled
//1 = Weak Pullup is enabled
PAADDR = 0x06;
PACTL = 0xff;

//Port A Alternate Function Enable (02): Reset value = 0x00
//0=Alternate Function Disabled
//1=Alternate Function Enabled
PAADDR = 0x02;                  //Select Alternate Function (Enable)
PACTL = 0x00;

//Set Port A Address register to point to NULL
PAADDR = 0x00;

/************************************************************/
//Port B Data Direction Register(01): Reset value = 0xff
//0=Output, 1=Input
PBADDR = 0x01;                  // Select PortB Data Dir Register
PBCTL = 0xf8;           // 1 1 1 1   1 0 0 0

//Port B Alternate Functions (07): Reset value = 0x00;
//Note: Since Maximum of 2 choices per pin, Port B AFS2 is not implemented
                        //PB0:  0=Reserved;    1=Ana0/AmpOut
                        //PB1:  0=Reserved;    1=Ana1/AmpInN
                        //PB2:  0=Reserved;    1=Ana2/AmpInP;
                        //PB3:  0=ClkIn;       1=Ana3
                        //PB4:  0=Reserved;    1=Ana7
                        //PB5:  0=Reserved;    1=Vref
                        //PB6:  0=Reserved;    1=Reserved
                        //PB7:  0=Reserved;    1=Reserved
PBADDR = 0x07;
PBCTL = 0x0f;

//Port B Output Control (03): Reset value = 0x00
//0 = Enable Active pullup for all output modes (Active Drain output)
//1 = Disable Active pullup (Open Drain output)
PBADDR = 0x03;
```

```
PBCTL = 0xf8;

//Port B High Drive Enable (0x04): Reset value = 0x00
//0 = High output drive disabled
//1 = High output drive enabled
PBADDR = 0x04;
PBCTL = 0x00;

//Port B STOPMODE recovery
//0 = Stop Mode Recovery is disabled for this bit
//1 = Stop Mode Recovery is enabled for this bit
PBADDR = 0x05;
PBCTL = 0x00;

//Port B Pullup Enable (0x06): Reset value = 0x00
//0 = Weak Pullup is disabled
//1 = Weak Pullup is enabled
PBADDR = 0x06;
PBCTL = 0xf8;

//Port B Alternate Function Enable (02): Reset value = 0x00
//0=Alternate Function Disabled
//1=Alternate Function Enabled
PBADDR = 0x02;        //Select Alternate Function (Enable)
PBCTL = 0x00;

//Set Port B Address register to point to NULL
PBADDR = 0x00;

/**************************************************************/
//Port C Data Direction Register(01): Reset value = 0xff
//0=Output, 1=Input
PCADDR = 0x01;                // Select PortC Data Dir Register
PCCTL = 0xf0;         // 1 1 1 1   0 0 0 0

//Port C Alternate Functions (07): Reset value = 0x00;
//Note: Since Maximum of 2 choices per pin, Port C AFS2 is not implemented
                      //PC0:  0=Reserved;   1=Ana4/CInP/LED
                      //PC1:  0=Reserved;   1=Ana5/CInN/LED
                      //PC2:  0=Reserved;   1=Ana6/LED;
                      //PC3:  0=ClkIn;      1=Cout
                      //PC4:  0=Reserved;   1=LED
                      //PC5:  0=Reserved;   1=LED
                      //PC6:  0=Reserved;   1=LED
                      //PC7:  0=Reserved;   1=LED
PCADDR = 0x07;
PCCTL = 0x0f;

//Port C Output Control (03): Reset value = 0x00
//0 = Enable Active pullup for all output modes (Active Drain output)
//1 = Disable Active pullup (Open Drain output)
PCADDR = 0x03;
```


OPTEK Technology Inc. — 1645 Wallace Drive, Carrollton, Texas 75006
Phone: (972) 323-2200 or (800) 341-4747 FAX: (972) 323-2396 sensors@optekinc.com www.optekinc.com

```
        PCCTL = 0xf0;

        //Port C High Drive Enable (0x04): Reset value = 0x00
        //0 = High output drive disabled
        //1 = High output drive enabled
        PCADDR = 0x04;
        PCCTL = 0x00;

        //Port C STOPMODE recovery
        //0 = Stop Mode Recovery is disabled for this bit
        //1 = Stop Mode Recovery is enabled for this bit
        PCADDR = 0x05;
        PCCTL = 0x00;

        //Port C Pullup Enable (0x06): Reset value = 0x00
        //0 = Weak Pullup is disabled
        //1 = Weak Pullup is enabled
        PCADDR = 0x06;
        PCCTL = 0xff;

        //Port C Alternate Function Enable (02): Reset value = 0x00
        //0=Alternate Function Disabled
        //1=Alternate Function Enabled
        PCADDR = 0x02;      //Select Alternate Function (Enable)
        PCCTL = 0x00;

        //Set Port C Address register to point to NULL
        PCADDR = 0x00;

        //Set up LED Driver output currents: Reset value = 0x00
        //0=disabled, 1=enabled
        LEDEN = 0x00;       // 0 0 0 0   0 0 0 0
        LEDLVLH = 0x00;     // 0,0=3 mA    0,1=7 mA    1,0=13 mA    1,1=20 mA
        LEDLVLL = 0x00;

        PAOUT = 0x00;       //All Outputs Low
        PBOUT = 0x00;       //All Outputs Low
        PCOUT = 0x00;       //All Outputs Low
}
```

Reference and Global Variables C-code referenced by Main Program

```
//refer.c
#include <eZ8.h>
#include "proto.h"

//unsigned short   u16ActivityCount; //inactive time counter

union
{
        struct
        {
                unsigned char tmh;
                unsigned char tml;
        } BSized;

        struct
        {
                unsigned short tmw;
        } WSized;
} uval;
```

Timing Initialization & Functions C-code referenced by Main Program

```
/***************************************************
*  Copyright (C) 2006 by OPTEK Technology
 *  All Rights Reserved
*  This code is provided as a tool for design engineers and OPTEK assumes no responsibility or liability.
 ***************************************************/

#include <eZ8.h>
#include "proto.h"
#include "refer.h"
#include "defines.h"

//////////////////////////////////////////////////////
//Intialize Timer-0
void init_timer0(void)
{
        //disable timer/counter
        T1CTL = 0x00;
//
        //T0CTL0 (0xf06) Timer/Counter Control Register0. Reset=0x00
        //Bits      7       6       5       4       3       2       1       0
        //Field     Tmode3  TICfg1  TICfg0  Reservd PWM2    PWM1    PWM0    InpCap
        //
        //T0CTL1 (0xf07) Timer/Counter Control Register1. Reset=0x00
        //Field     TEn     TPol    Pres2   Pres1   Pres0   Tmode2  Tmode1  Tmode0
        //
        //Tmode3210     0000=One-Shot   0001=Continuous         0010=Counter    0011=PWM Sngl Output
        //              0100=Capture    0101=Compare            0110=Gated      0111=Capture/Compare
        //              1000=PWM DualOut1001=Capture/Restrt     1010=Comparator Counter
        //TICfg10       0x=Interrupt occurs on all defined Reload, Compare and Input events
        //              10=Interrupt only on defined Input Capture/Deassertion Events
        //              11=Interrupt only on defined Reload/Compare Events
```

OPTEK reserves the right to make changes at any time in order to improve design and to supply the best product possible.

OPTEK Technology Inc. — 1645 Wallace Drive, Carrollton, Texas 75006
Phone: (972) 323-2200 or (800) 341-4747 FAX: (972) 323-2396 sensors@optekinc.com www.optekinc.com

```
	//Reserved		Must be 0
	//PWMD210		PWM Delay Value
	//			000=No Delay	001=2 cycles	010=4 cycles	011=8 cycles
	//			100=16 cycles	101=32 cycles	110=64 cycles	111=128 cycles
	//InpCap Input Capture Event (read only)
	//			0=Previous timer Interrupt is not from Timer Input Capture Event
	//			1=Previous timer Interrupt is from Timer Input Capture Event
	//
	//TEn			0 = Timer is disabled, 1 = Timer is enabled to count
	//TPol			Meaning of this bit is dependent on Tmode (Timer Mode)
	//Pres210		Prescale Value: 000=1	001=2	010=4	011=8
	//				100=16	101=32	110=64	111=128


	T0CTL0 = 0x00;
	T0CTL1 = 0x02;

	//T0H (0xf00) & T0L (0xf01) Current timer/counter value. Reset=0x0001
	// timer/counter is an Up counter.  A read of T0H will put T0L into a temporary
	// holding register so that it can be read.
	T0H		= 0x00;	// Timer High
	T0L	= 0x01;	// Timer Low

	//T0RH (0xf02) & T0RL (0xf03) Point at which counter/timer reloads to 0x0001. Reset=0xffff
	T0RH	= 0xFF;	// Reload Compare High
	T0RL	= 0xFF;	// Reload Compare Low

	// Enable Timer/Counter mode
	T0CTL1 = T0CTL1 |= 0x80;


	//T0CTL0 (0xf06) & T0CTL1 (0xf07) Timer Control Registers. Reset=0x0000
}

/************************************************************
* Delmsec delays by MsecCnt milliseconds
************************************************************/
void Delmsec(unsigned short MsecCnt)
{
	while (MsecCnt --)
	{
		Del1msec();
	}
}

/************************************************************
* 1 millisecond delay routine
* Note: Delay value is 32 bit unsigned number
************************************************************/
void Del1msec(void)
{
	// 0.001Sec/5.5MHz=5500 Cycle delay for 1 mSec
	asm("MSEC_COUNT	.set (275)");		// (5500-23)/18=304
						// FetchCycles+InstructionCycles Ftotal Itotal Sum
	asm("\tpush R12");			// 2+2  2  2  4
	asm("\tpush R13");			// 2+2	4	4	8
	asm("\tLD   R12, #HIGH(MSEC_COUNT)");	// 3+2	7	6	13
   asm("\tLD   R13, #LOW(MSEC_COUNT)");		// 3+2	10	8	18
```

```
        asm("$msecCountDown:");                               //this loop takes about 36 cycles"
        asm("\tCP   R13,#%0");                                // 4+3   4      3      7
        asm("\tCPC  R12,#%0");                                // 4+3   8      6      14
   asm("\tJR   ule,$msecdelay_done");              // 2+2   10     8      18
        asm("\tSUB  R13,#%1");                                // 3+3   13     11     24
   asm("\tSBC  R12,#%0");                          // 3+3   16     14     30
        asm("\tjr $msecCountDown");                           // 2+2   18     16     34

        asm("$msecdelay_done:");
        asm("\tpop R13");                                     // 3+2   13     10     23
        asm("\tpop R12");                                     // 3+2   16     12     28
        //asm("\tret");                                              // 1+4   17     16     33
}

/****************************************************************
ReadTimer0
****************************************************************/
void ReadTimer0(void)
{
        uval.BSized.tmh = T0H;
        uval.BSized.tml = T0L;
}


/***************************************************************
SetTimer0
***************************************************************/
void SetTimer0(void)
{
        //T0H (0xf00) & T0L (0xf01) Current timer/counter value. Reset=0x0001
        // timer/counter is an Up counter.  A read of T0H will put T0L into a temporary
        // holding register so that it can be read.
        T0H     = 0x00;  // Timer High
        T0L     = 0x01;  // Timer Low
}
```

Ptototype Definitions—Header Included File C-code referenced by Main Program

```
/*************************************************
*  Copyright (C) 2006 by OPTEK Technology
 *  All Rights Reserved
*  This code is provided as a tool for design engineers and OPTEK assumes no responsibility or liability.
 *************************************************/

#ifndef PROTO
#define PROTO

// gpio.c
void init_gpio(void);

// main.c
void main(void);
void PortBBitsOn(unsigned char);
void PortBBitsOff(unsigned char);
```


OPTEK Technology Inc. — 1645 Wallace Drive, Carrollton, Texas 75006
Phone: (972) 323-2200 or (800) 341-4747 FAX: (972) 323-2396 sensors@optekinc.com www.optekinc.com

```
// timer.c
void Del1msec(void);
void Delmsec(unsigned short);
void init_systemclock(void);
void init_timer0(void);
void ReadTimer0(void);
void SetTimer0(void);

#endif
```

Ptototype Definitions—Header Included File C-code referenced by Main Program

```
/extern union
{
	struct
	{
		unsigned char tmh;
		unsigned char tml;
	} BSized;

	struct
	{
		unsigned short tmw;
	} WSized;
} uval;
```

End of Code