BM017 Color Sensor

I2C Color Sensor with White LED for Illumination

hardware made easy

Solutions Cubed

designservices@solutions-cubed.com
phone – 530.891.8045
256 East First Street
Chico, CA 95928

Contact Solutions Cubed, LLC for your custom designs:

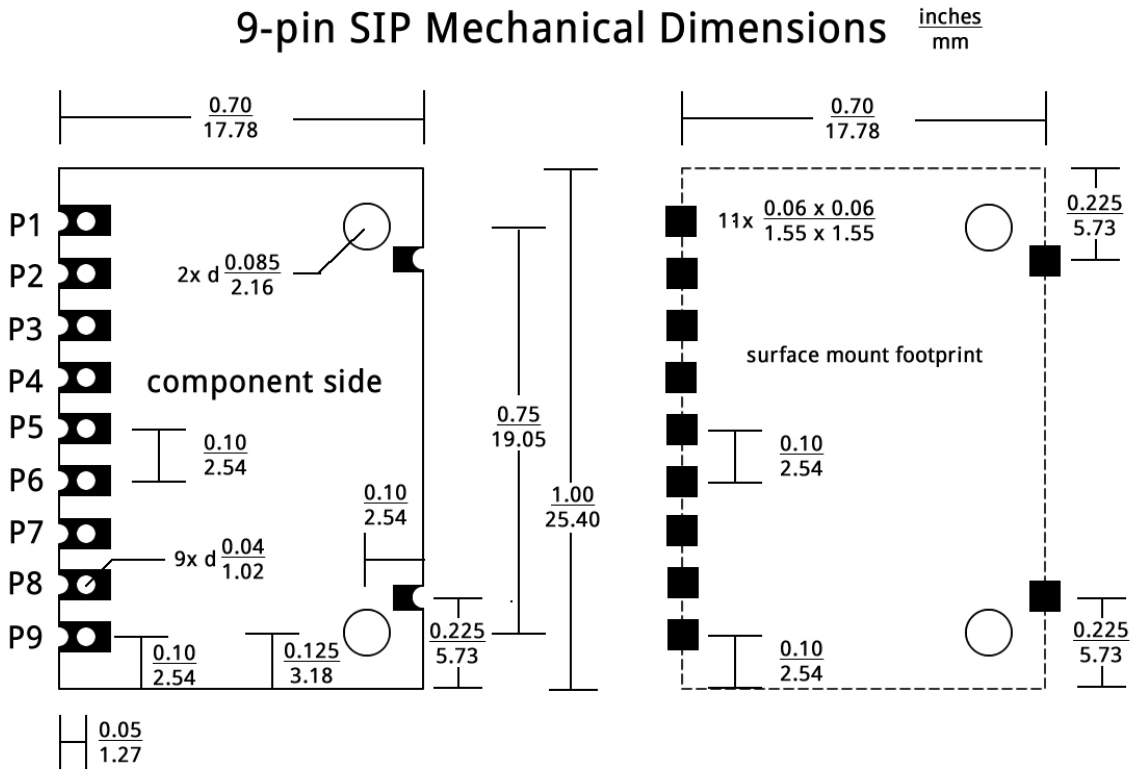
Solutions Cubed is an innovative electronic design firm. We have created successful designs for a myriad of industries including mass produced consumer products, deep-sea robotic components, and encrypted encoders for the banking industry. We love meeting new customers and are interested in hearing about your design needs.

Product Description:

The BM017 module carries one AMS TCS34725 color light-to-digital converter IC. The IC has an I2C interface allowing digital systems to measure object color. Red, blue, green, and clear values may be read. The module also has a white LED and MOSFET that may be used to illuminate objects. Voltage level translation circuitry is included allowing the module to interface to 3V or 5V logic systems. A 3.3V linear regulator on the module provides power to the color sensor.

- Red, blue, green, and clear digital color measurements
- White LED on board may be used to illuminate objects being tested
- On-board 3.3V regulator
- Voltage level translation circuitry allows use with 3V or 5V logic
- I2C interface

Dimensions:



Specifications:

Characteristic	Min	Typ	Max	Unit	Notes
VIN Operating voltage	4		20	V	Input to 3.3V regulator
VDD Voltage		3.3V			
Operating current		2		mA	
SCL and SDA pull-up resistors		4700		Ω	Pulled to VDD_IO
Color conversion time	2.4		700	mS	This is configured via the I2C interface
Color count	1024		65535		Longer conversion time provides higher counts
Operating temperature	-30		+70	°C	

Pin Functions and Notes

#	Name	Maximum Voltage	Notes
1	VIN	20VDC	Input voltage for on-board 3.3V regulator
2	GND	0V	Ground return for module
3	VDD	3.3V	Normally the output for the 3.3V regulator, can be used as an input to supply 3.3V to the module
4	VDD_IO	5V	Sets the voltage for the I2C interface. If your system is a 5V system apply 5V here, if it is a 3.3V system then connect VDD or another 3.3V source.
5	SCL	VDD_IO	I2C clock line
6	SDA	VDD_IO	I2C data line
7	INT	3.3V	Open-drain output asserted low for various interrupt settings. Pull to 3.3V externally with a 4.7KΩ resistor is used.
8	VDD_LED	3.3V(5V)	Power supply for white illuminating LED. There is a 47Ω resistor in series with the LED. If powering with a 5V supply you should add a second 47Ω external to the module to ensure the input sinks less than 20mA.
9	LEDON	5V	Gate connection to N-channel MOSFET that turns on illuminating LED.

User Notes/Tips

1. Color measurements can be dependent on illumination, the proximity of the object being measured, and the angle of measurement. Ideally you want a well-lit object relatively close to the sensor and parallel to the face of the sensor.

   You can use look-up tables to calibrate for the light source you are using for illumination. You can also use relative measurements to determine colors. For example if red is greater than green and blue, then the object is likely red.

Communication Protocol

Hardware Usage:

The BM017 uses an I2C interface to write/read data to/from the TCS34725 color sensor. The color sensor's 7-bit address is 0x29. The I2C interface follows the standard I2C format. You should review the datasheet for the TCS34725 for more details on the communication protocol and various register functions in the color sensor. This datasheet will only cover basic information.

S – start condition W – write bit (0) R - read bit(1) A – acknowledge(0)
P – stop condition

Master-to-slave
Slave-to-master

I2C Write Format

1	7	1	1	8	1	8	1	?	1
S	Slave address	W	A	Command code	A	Data byte	A	Continue if multiple bytes	P

I2C Read Format

1	7	1	1	8	1	8	1	?	1
S	Slave address	R	A	Data byte	A	Data byte	A	Continue if multiple bytes	P

I2C Combined Read Format (uses a write to establish register to read from)

1	7	1	1	8	1	1	7	1	1	8	1	1
S	Slave address	W	A	Command code	A	S	Slave adddress	R	A	Data	A	P

Command Byte: The command byte is used to set the register address of a write or read operation.

Command Byte – in this example would be B'10110100' or 0xB4 (hexadecimal)	
Bits 7:5	Bits 4:0
Normally B'101' see datasheet for more info	Address index value. For example the first color address CDATAL is B'10100'. See the table that follows for the address values of other registers.

Example Reading from CDATAL register

1	Slave address and write bit 0x29 in bits 7:1 and 0 in bit 0	1	Command code includes CDATAL address	1	1	Slave address and read bit 0x29 in bits 7:1 and 1 in bit 0	1	CDATAL contents returned	1	1
S	0x52	A	0xB4	A	S	0x53	A	Data	A	P

Register Definitions: The TCS34725 registers are shown here for reference. For additional details review the TCS34725 datasheet.

Address	Register Name	R/W	Register Function
--	COMMAND	W	Used to specify register addresses
0X00	ENABLE	R/W	Enables the color sensor
0X01	ATIME	R/W	Conversion integration timer
0X03	WTIME	R/W	Wait timer
0X04	AILTL	R/W	Clear interrupt low threshold low byte
0X05	AILTH	R/W	Clear interrupt low threshold high byte
0X06	AIHTL	R/W	Clear interrupt high threshold low byte
0X07	AIHTH	R/W	Clear interrupt high threshold high byte
0X0C	PERS	R/W	Interrupt persistence filter
0X0D	CONFIG	R/W	Configuration byte
0X0F	CONTROL	R/W	Control byte
0X12	ID	R	Device ID
0X13	STATUS	R	Device status
0X14	CDATAL	R	Clear data low byte
0X15	CDATAH	R	Clear data high byte
0X16	RDATAL	R	Red data low byte
0X17	RDATAH	R	Red data high byte
0X18	GDATAL	R	Green data low byte
0X19	GDATAH	R	Green data high byte
0X1A	BDATAL	R	Blue data low byte
0X1B	BDATAH	R	Blue data high byte

Pseudo Code for Initializing Sensor:
To initialize the sensor for operation you need to set the analog integration time in the ATIME register, set the gain on the analog conversion in the CONTROL register, and enable the oscillator and analog converters in the ENABLE register. Color data may then be read from the sensor.

```
#define EnableAddress 0xa0 // register address + command bits
#define ATimeAddress 0xa1 // register address + command bits
#define ControlAddress 0xaf // register address + command bits

void init_TCS34725(void)
{
  i2cWriteBuffer[0] = 0xF6;
  Writei2cRegisters(1,ATimeAddress);    // RGBC timing is 256 - contents x 2.4mS =
  i2cWriteBuffer[0] = 0x00;
  Writei2cRegisters(1,ControlAddress);  // RGBC gain control 0x00 is gain of 1
  i2cWriteBuffer[0] = 0x03;
  Writei2cRegisters(1,EnableAddress);    // enable ADs and oscillator for sensor
}
```

Example Code for Reading and Writing: These routines for the Arduino can be used as examples for reading and writing data to the TCS34725. They use functions from the "Wire" library.

```
/*
Load the i2cWriteBuffer with the data you want to write and then call this routine with appropriate address and
command bits and the number of bytes you are writing.
*/
void Writei2cRegisters(byte numberbytes, byte command)
{
   byte i = 0;

   Wire.beginTransmission(SensorAddressWrite);   // Send address with Write bit set
   Wire.write(command);                          // Send command, normally the register address
   for (i=0;i<numberbytes;i++)                   // Send data
     Wire.write(i2cWriteBuffer[i]);
   Wire.endTransmission();

   delayMicroseconds(100);      // allow some time for bus to settle
}


/*
Send register address to this function the number of bytes you want to read and it returns the byte value for the
register's contents in the i2cReadBuffer
*/
byte Readi2cRegisters(int numberbytes, byte command)
{
  byte i = 0;

   Wire.beginTransmission(SensorAddressWrite);   // Write address of read to sensor
   Wire.write(command);
   Wire.endTransmission();

   delayMicroseconds(100);      // allow some time for bus to settle

   Wire.requestFrom(SensorAddressRead,numberbytes);   // read data
   for(i=0;i<numberbytes;i++)
     i2cReadBuffer[i] = Wire.read();
   Wire.endTransmission();

   delayMicroseconds(100);      // allow some time for bus to settle
}

#define ColorAddress 0xb4 // register address + command bits

/*
Reads 8 bytes from color registers and converts them to 16-bit color measurements
*/
void get_Colors(void)
{
  unsigned int clear_color = 0;
  unsigned int red_color = 0;
  unsigned int green_color = 0;
  unsigned int blue_color = 0;

  Readi2cRegisters(8,ColorAddress);
  clear_color = (unsigned int)(i2cReadBuffer[1]<<8) + (unsigned int)i2cReadBuffer[0];
  red_color = (unsigned int)(i2cReadBuffer[3]<<8) + (unsigned int)i2cReadBuffer[2];
  green_color = (unsigned int)(i2cReadBuffer[5]<<8) + (unsigned int)i2cReadBuffer[4];
  blue_color = (unsigned int)(i2cReadBuffer[7]<<8) + (unsigned int)i2cReadBuffer[6];
}
```

Register Details:
Some registers are defined here for reference. See the TCS34725 datasheet for more details.

ENABLE Register (0x00):
Used to enable the sensor color conversions.

Field	Bits	Description
reserved	7:5	Write as 0
AIEN	4	When set RGBC interrupts are enabled
WEN	3	When set the wait timer is activated
reserved	2	Write as 0
AEN	1	When set color conversions occur
PON	0	When set the internal oscillator is running (conversions can occur)

ATIME Register (0x01):
Sets the integration/conversion time. Example settings are shown below.
Max Count = (256-ATIME) x 2.4mS

Register Value	Time	Max Count (loaded into color registers)
0xFF	2.4mS	1024
0xF6	24mS	10240
0xD5	103mS	44032
0xC0	154mS	65535
0x00	614mS	65535

CONTROL Register (0x0F):
The two lowest bits of the CONTROL register are used to set the gain for the analog conversions. Bits 7:2 are reserved and should be written as 0.

Field	Bits	Description
reserved	7:2	Write as 0
AGAIN	1:0	00 1x gain 01 4x gain 10 16x gain 11 60x gain

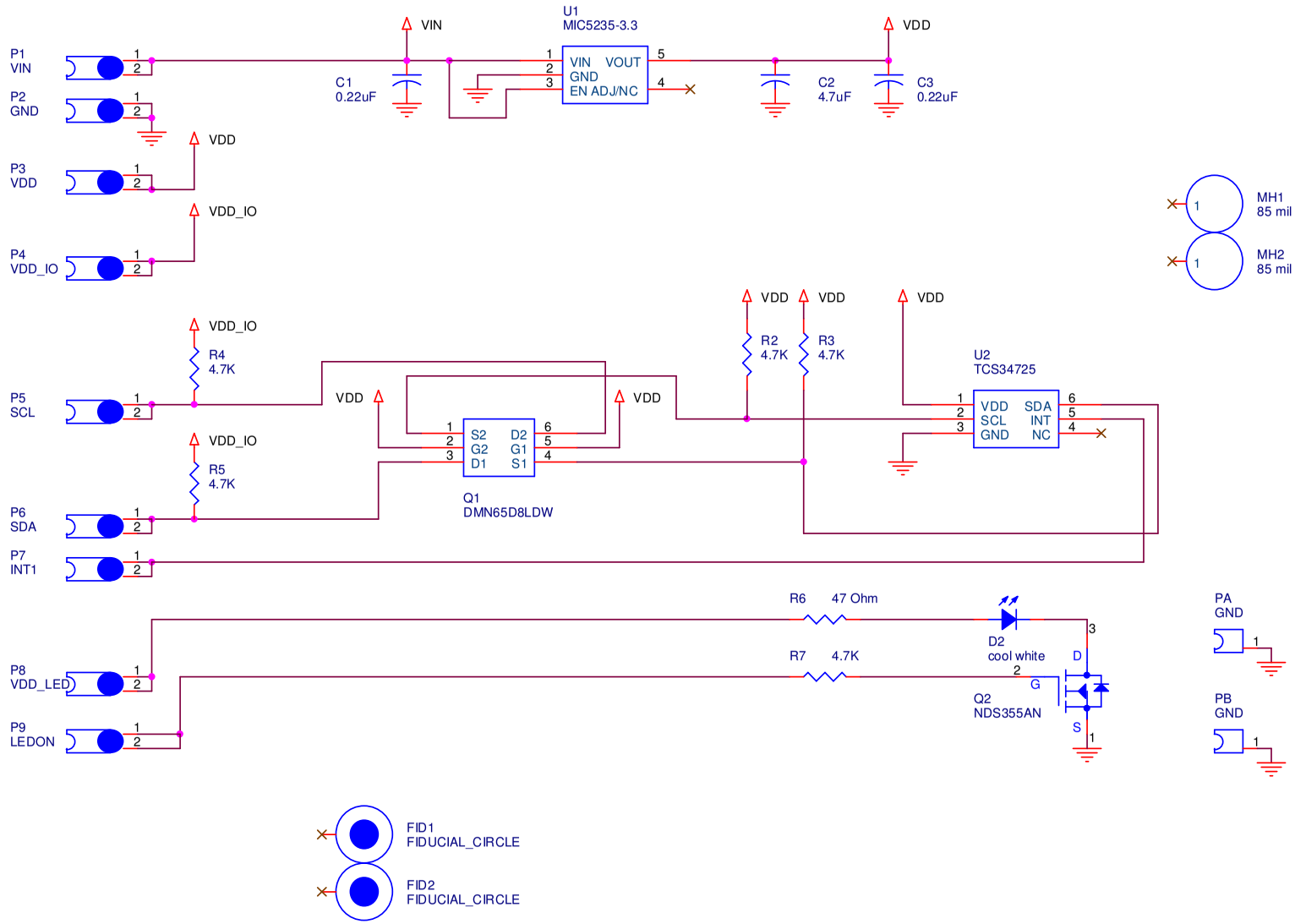
Color Registers (0x14 through 0x1B):
Color registers are set up as pairs (clear, red, green, and blue) each with a high byte and a low byte. The color measurement can be calculated by

Color = xDATAH * 256 + xDATAL

To convert it to a 10-bit measurement use…

Color = (xDATAH * 256 + xDATAL) / (256-ATIME)

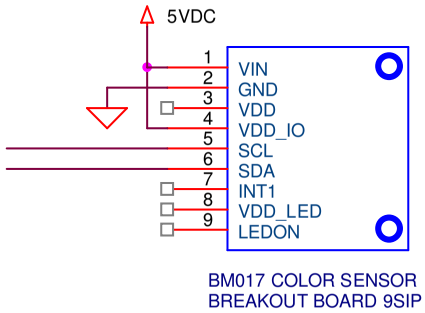
Schematics:


P1
VIN
P2
GND
P3
VDD
P4
VDD_IO
P5
SCL
P6
SDA
P7
INT1
P8
VDD_LED
P9
LEDON
VIN
U1
MIC5235-3.3
VIN VOUT
GND
EN ADJ/NC
C1
0.22uF
C2
4.7uF
C3
0.22uF
VDD
VDD_IO
R4
4.7K
R5
4.7K
R2
4.7K
R3
4.7K
U2
TCS34725
VDD SDA
SCL INT
GND NC
S2 D2
G2 G1
D1 S1
Q1
DMN65D8LDW
R6 47 Ohm
R7 4.7K
D2
cool white
Q2
NDS355AN
D
G
S
MH1
85 mil
MH2
85 mil
PA
GND
PB
GND
FID1
FIDUCIAL_CIRCLE
FID2
FIDUCIAL_CIRCLE

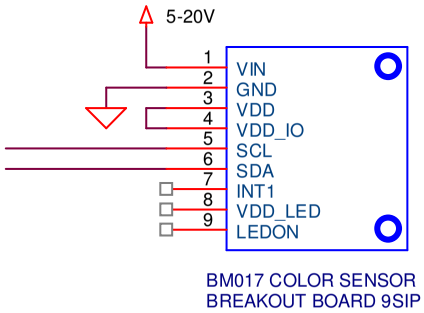
Application Schematics:

POWER OPTIONS

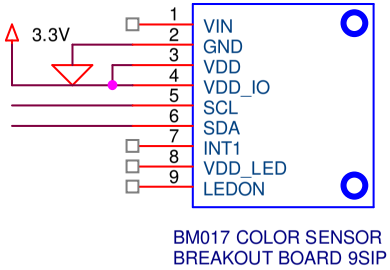
5V SYSTEM



3.3V SYSTEM

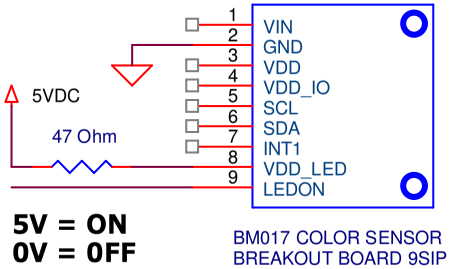


ALT. 3.3V SYSTEM



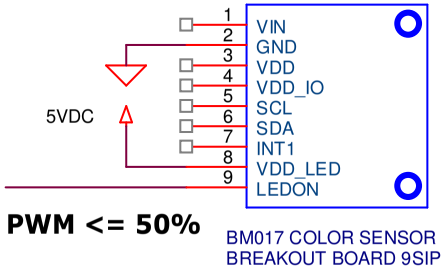
ILLUMINATION LED OPTIONS

5V SYSTEM
20mA LED CURRENT



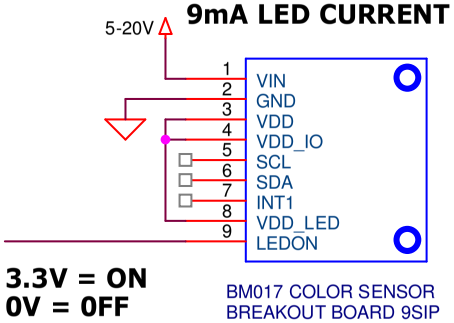
5V = ON
0V = 0FF

5V SYSTEM
20mA LED CURRENT



PWM <= 50%

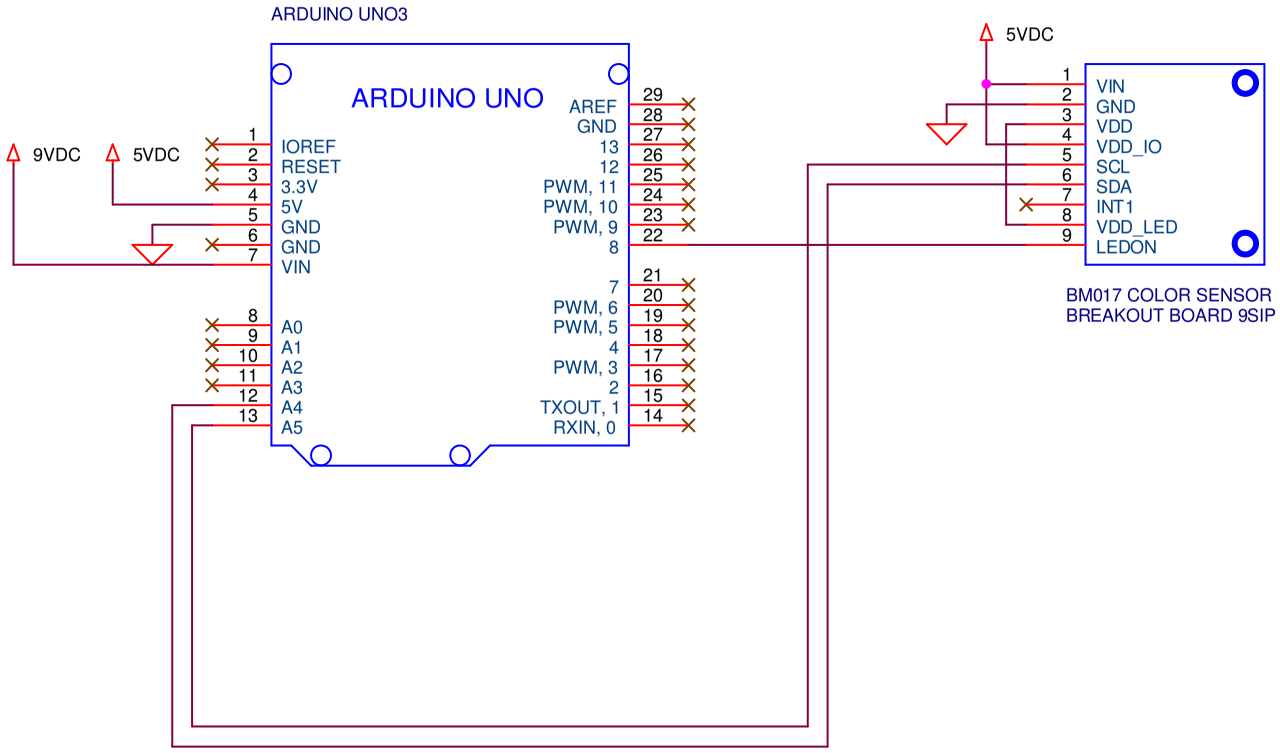
3.3V SYSTEM
9mA LED CURRENT



3.3V = ON
0V = 0FF

Application Schematics:

ARDUINO UNO CONNECTIONS



Application Schematics:

COLOR SENSOR TO RGB LED