

# RGBSPCHIP5

## Vista 5

# Operating Manual

### **Applicable Products:**

RGBSPCHIP5 chip

RGBSPVista5 pre-made assembly and kit

### **Introduction**

This manual describes how to operate RGBSP 5 series products, and to use the RGBSP 5 series products in electrical environments, including circuits you may design of your own.

In some sections of this manual there may be a bit of technical detail. RGBsunset Productions is firmly of the belief that users should have as much information on their purchase as possible, and get very annoyed at ‘manuals’ for other products that over simplify things. Your RGBSP Vista 5 or RGBSPCHIP5 product has a variety of capabilities and knowing them will help you better appreciate your purchase.

It’s more than just a light.

If you’d rather get straight into using the product, jump to the Getting Started section. When you feel like reading the manual a bit more closely, the manual will indicate which sections provide technical details that can be skipped.

Your RGBsunset Productions 5 series products including an RGBSPVista5 assembly or a single RGBSPCHIP5 are designed to operate a large number of red/green and blue LEDs to create interesting colour patterns. They can be controlled by an Infra-Red remote or a simple serial protocol. The 5 series creates an individual brightness level for 15 independent channels. These are usually grouped into 5 individual channels of red/green and blue. By combining these colours a variety of patterns can be created. Individual channels can be controlled by the serial protocol. RGBSP Vista5 is a PCB designed to be incorporated into lighting housings or sculptures to provide lighting patterns or colours. The RGBSPCHIP5 is the heart of the RGBSP Vista5 product, and is available from RGBsunset Productions for experienced users wanting to create their own LED lighting experiments. OEMs are encouraged to contact RGBsunset Productions if they wish to use the RGBSPCHIP5 in their own commercial products.

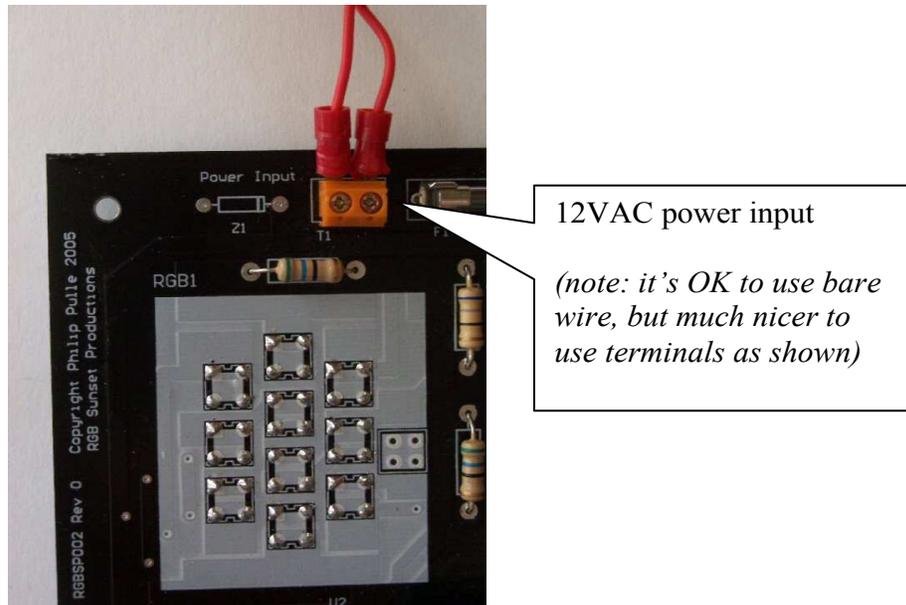
## Table of Contents

Applicable Products:.....	1
Introduction.....	1
Connecting your unit to power .....	3
Connecting the infra-red receiver.....	3
Infra Red Remote .....	4
Infra Red Transmission Physical Details.....	5
Programmatic Control of the unit .....	6
RGBSPCHIP5 or RGBSPVista5 programmatic control command set .....	7
CMD_EXT.....	8
CMD_SETALL.....	8
CMD_SETGROUP .....	8
CMD_MOVEGROUP.....	8
CMD_ALLCOLOUR.....	8
CMD_ALLMOVE .....	8
CMD_SETSPEED.....	9
CMD_SETMODE.....	9
CMD_IRKEY.....	9
CMD_SETPARM.....	10
CMD_SETLED.....	10
CMD_MOVELED .....	10
CMD_SETPRESET.....	10
CMD_SETANNUN.....	11
CMD_SETPINOUT.....	11
Programmatic Control Examples.....	12
Suggestions on Coding an Algorithm to transmit the Manchester Code .....	12
Infra Red Receiver Circuitry.....	14
Determining Remote Compatibility.....	14
Imprinting your remote.....	15
IR Remote Keys and Operation.....	16
On/Standby Pattern .....	17
RGB Maximum setting .....	18
Creating colours .....	18
Creating a white colour .....	19
Index .....	20

### **Connecting your unit to power**

Your RGBSPVista5 kit or pre-made assembly simply needs to be connected to a 12V AC power source. A common halogen lighting transformer can be used; however ‘solid state’ or dimmable halogen power supply **should not** be used because they condition the signal in a way that does not supply a true 12VAC power signal.

The 12V AC power signal should connect via the terminal block in the position shown below:

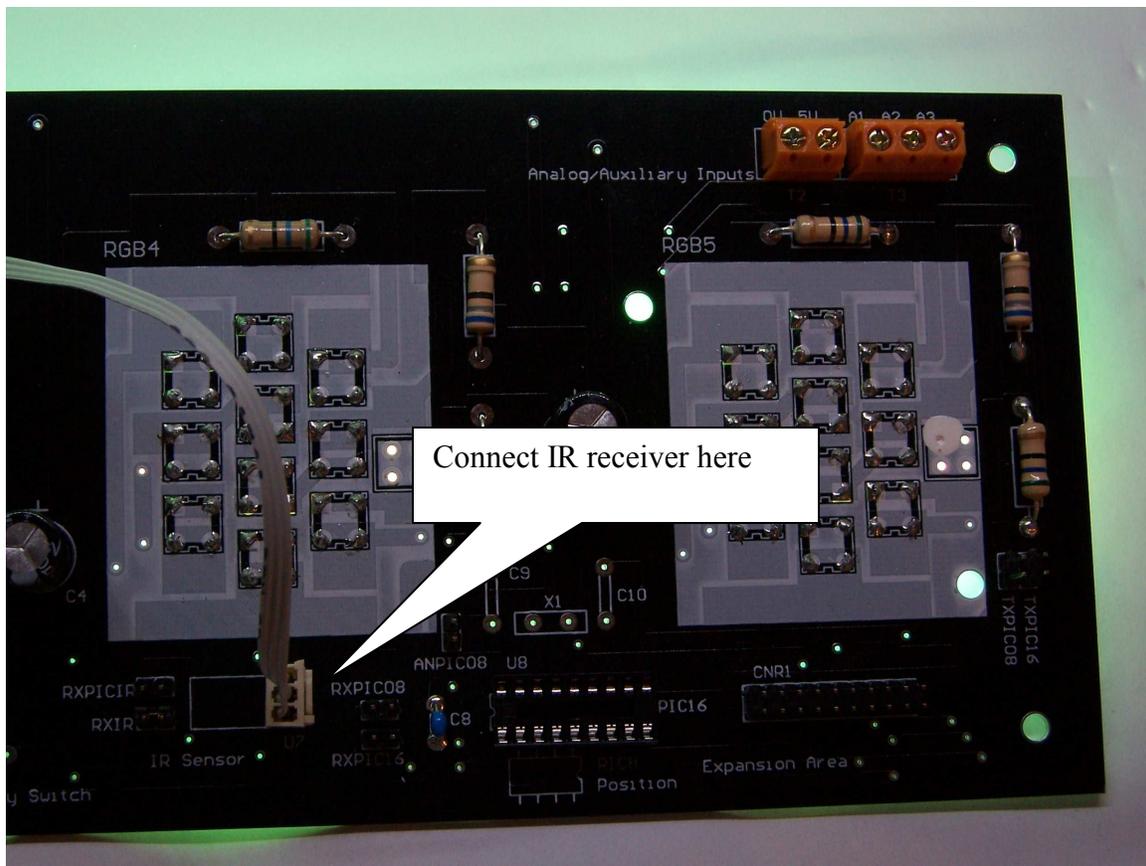


### **Connecting the infra-red receiver**

If you have a pre-built or pre-housed or kit unit you should have been supplied with a small board or box with a three pin lead. This is the infra-red receiver and contains a sensor for detecting signals from the infra-red remote control plus a couple of other components to filter the power signal from stray pulses.

This receiver should be placed in a position away from direct illumination from the LED lighting unit and any other powerful lights or else sensitivity may suffer.

The receiver should be connected to the unit in the position below:



## ***Infra Red Remote***

This section details how to control light patterns in RGBSP 5 Series products using a common remote control. You may use most cheap programmable remotes you might find in a supermarket or an old unused remote (of certain kinds) or you might even be able to use your own existing programmable remote, allowing you to control not only your home entertainment appliances but your RGBsunset lighting products in one unit.

The RGBSP 5 Series products use a commonly known scheme called RC5, originally developed by the Philips© Corporation for its equipment. Nearly all programmable remote controls can be programmed to mimic a Philips© remote control. Users should consult the manual that came with their remote control to figure out how to setup their remote for use with RGBSP 5 Series units. The Determining Remote Compatibility section has some pointers on how to get your remote working, though because of the large variety of different remote controls on the market specific details are impossible.

*If you have a pre-made assembly or kit you may wish to skip the next two sections which are focused to RGBSPCHIP5 users wishing to design their own infra-red receiver circuitry, and go straight to the Determining Remote Compatibility and IR Remote Keys and Operation section. You can come back to this section if you wish to learn a little more about how infra-red remote controls work though.*

## Infra Red Transmission Physical Details

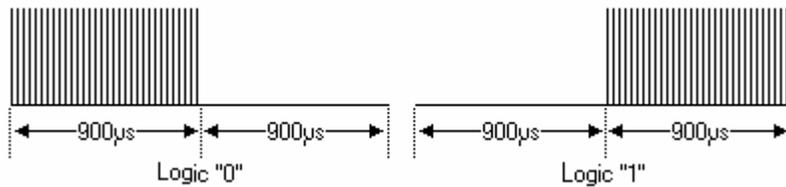
The infra red remote control used by most home entertainment appliances, including programmable remotes, usually emit an infra red signal of about 36kHz to 44kHz carrier frequency. A carrier is an alternating on/off signal at the carrier frequency. Different manufacturers use a variety of signaling schemes, or on/off sequences of these signals. RGBsunset products are designed to receive signals using the RC5 scheme. RC5 signals consist of 14 bits, first two start bits, a toggle bit, 5 address bits and 6 command bits.

- The start bits allow a receiver to lock onto the signal
- The toggle bit is used to determine if the user has just pressed a button, or the signal is a repeat because they have held the button down.
- The 5 bit address identifies what sort of unit the remote is controlling. For example a VCR may have a different address to a CD player. Both use the RC5 signaling system, but only one responds at a time.
- The 6 bit command actually identifies the button that was pressed or is being held down.

Each bit is approximately 1.8mS long, meaning a button press results in  $(14 \times 1.8\text{mS}) = 25.2\text{mS}$  set of pulses from the remote.

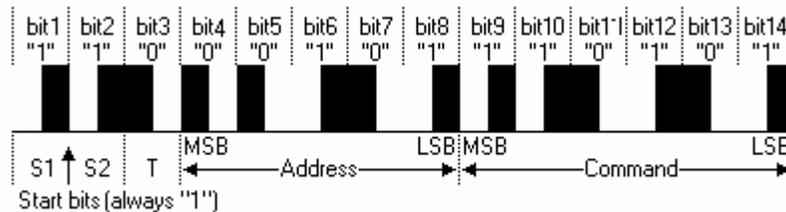
The bits are coded in a method called ‘Manchester Coding’ or ‘Bi-Phase Coding’. In this scheme each bit is split into two 900uS halves. When logic ‘0’ bit is sent, carrier signal is sent for 900uS, and then no carrier is sent for the other half of the bit time. When logic ‘1’ bit is sent, there is no carrier signal for 900uS, and then in the remaining half the carrier is sent.

The image below shows this:



The sharp spikes in this diagram show the 36kHz-44kHz carrier signal, that is, when infra-red light is actually emitted from the remote control. The flat areas show when there is no infra-red radiation being emitted from the remote control.

The diagram below shows how a RC5 code of bit pattern 11000101110101 would be sent.



In this case the toggle bit is clear, the address is 0x05(hex) and the command bit is 0x35. From the RC5 protocol documentation, this would be the Play button for a VCR device. See references below on the internet for more information on the RC5 protocol.

- reference 1
- reference 2

It may seem strange to go to the effort of splitting the bits of a message in this manner. However Manchester coding has the advantage that a receiver can lock into the timing of the signal using the first two start bits, thus allowing for variances in the timing of different remote control devices. Also there is some measure of error detection, in that a noisy environment may inadvertently inject or suppress a signal during either half of a bit. If the receiver sees two highs or two lows in a bit time instead of a high-low/low-high transition, then it can ignore that transmission.

Note: The RC6 protocol extends RC5 with an extra command bit. The still remains at 14 bits however to retain compatibility with RC5 receivers. The extra bit is inverted and transmitted as the second start bit. This product is compatible with RC6 remotes because it only uses commands where the extra bit is zero '0', hence transmitted as S2='1'.

## ***Programmatic Control of the unit***

The RGBSPVista 5 or RGBSPCHIP5 can be controlled manually via a remote control, which is applicable for interactive situations...eg sitting on the couch watching TV!

However it can be additionally controlled by your own microcontroller projects! You may wish to check out the 'Experimenter Section' topic which details an area on the RGBSPVista5 board that is especially reserved for hobbyists who wish to extend the operation of their unit.

Programmatic control is via a variation of the RC5 protocol described above. A command to the unit consists of a packet of data bytes sent at the same timing and the same Manchester coding scheme however with only 10 bits instead of the above 14 bits. The first two bits are the start bits which are used to synchronize timing and the remaining eight bits are the data.

When the unit receives data on the rx pin that is coded with Manchester coding, if it is 10 bits long it assumes it is a programmatic control byte. If it is 14 bits long it assumes it is from a remote control. If it is neither, it assumes an error and the data are discarded.

Each packet consists of a command byte, a variable number of data bytes (depending on the command) and a checksum.

Command	Data 0	Data 1..n-1	Data n	Checksum
---------	--------	-------------	--------	----------

The checksum sum is simply the value 0xff XOR'd with every byte in the packet excluding the checksum.

The table below shows the commands that can be sent to the RGBSPVista5 or RGBSPCHIP5 unit.

### RGBSPCHIP5 or RGBSPVista5 programmatic control command set

Command		Data	Description
Name	Value		
CMD_EXT	0x13	D0,D1,D2	<i>Reserved for future implementation</i>
CMD_SETALL	0x2f	r1,g1,b1,r2,g2,b2,r3,g3,b3,r4,g4,b4,r5,g5,b5	Set all leds to a value
CMD_SETGROUP	0x34	group,r,g,b	Set a group to a rgb colour
CMD_MOVEGROUP	0x44	group,r,g,b	Move a group to a rgb colour
CMD_ALLCOLOUR	0x53	r,g,b	Set all groups to a rgb colour
CMD_ALLMOVE	0x63	r,g,b	Move all groups to a rgb colour
CMD_SETSPEED	0x71	Speed := {0..7}	Set the speed of patterns
CMD_SETMODE	0x81	Mode := {0..4}	Set the pattern mode
CMD_IRKEY	0x91	irkey	Simulate an ir remote key press
CMD_SETPARM	0xa3	max_red,max_green,max_blue	Set the max rgb for patterns
CMD_SETLED	0xb2	led,value	Set an individual led value
CMD_MOVELED	0xc2	led,value	Move an individual led to a value
CMD_SETPRESET	0xd4	preset,r,g,b	Set a preset to an rgb colour
CMD_SETANNUN	0xe1	Annunciator flags	Set Annunciator flags
CMD_SETPINOUT	0xff	p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15	Change the pinout for the groups

Some notes before getting into each command in detail:

- The second nibble of the command is the number of bytes of data for the command. This may be useful to remember in programming your code to interface to the chip
- Each byte in a command packet must be transmitted/received within 1 byte time or else the entire packet is discarded and the device waits for a time before receiving any more data.
- Some commands like CMD\_SETPRESET,CMD\_SETANNUN and CMD\_SETPINOUT can alter the operation of the chip so that it will be difficult to operate with a remote control. The CMD\_SETANNUN command can be used to reset the chip to defaults.

- As the name suggests, the RGBSPCHIP5 (used in RGBSPVista5) controls five RGB channels, so r1,g1,b1 is the rgb setting for the first group r2,g2,b2 is the second group and so on. Each r,g,b value can range from 0..255 (off to full intensity)
- Each group has a different led. See the pinout diagram for a description of pinout assignment and the CMD\_SETPINOUT to change it as desired for your application
- Value ranges are as below. If a value is transmitted out of the allowed range, unpredictable results may occur

```
group      := {0...4}
r,g,b     := {0...255}
preset    := {0...15}
led       := {0...4, 6...15} /*note led 5 is unused*/
value     := {0..255}
max_red,
max_green, := {0...255}
max_blue
speed     := {0..7}
mode      := {0..4}
flags     := {bits have specific functions, see description}
p1..p15  := {0..15}
```

### ***CMD\_EXT***

This command is currently unimplemented and reserved for future use.

### ***CMD\_SETALL***

Change the mode to fixed colour, and set each groups rgb setting individually

### ***CMD\_SETGROUP***

Set an rgb channel to a specified rgb colour.

### ***CMD\_MOVEGROUP***

Move the group specified to a given rgb colour. The speed to the move is defined by the current speed setting.

### ***CMD\_ALLCOLOUR***

Set all groups to the specified colour

### ***CMD\_ALLMOVE***

Move all groups to the specified rgb colour at the current speed setting

### ***CMD\_SETSPEED***

The speed for all ‘move’ commands and dynamic patterns. There are eight speeds, 0 is the fastest speed and 7 is the slowest speed. Some patterns look better at slow speeds, others look better at fast speeds.

### ***CMD\_SETMODE***

There are 5 pattern modes:

0	Lava	Colours are generated at one end and drip down to the other end. The end that generates the colour alternates between each speed setting.
1	Random	Each group randomly moves between colours
2	Bar	All groups randomly move between colours
3	Blinky	All groups randomly pick a colour without smoothly moving between them. Annoying but attention grabbing.
4	Fixed	No movement, the colour of all groups can be changed by remote or CMD_SETPARM

### ***CMD\_IRKEY***

This command simulates pressing an IR key on a remote control. The data field defines the RC5 command. The RC5 commands that the unit will respond to are

RC5 Command	Value	RGBSP Command	Description
On/Standby	0x0c	On/Standby	Turn the unit on or off. Note in off state, the unit is still powered but will only have all LEDs off and will only respond to another On/Standby command.
Chan -	0x21	Speed faster	Make patterns go faster
Chan +	0x20	Speed slower	Make patterns go slower
Vol -	0x11	Previous Pattern	Select previous pattern
Rewind	0x32		
Vol-	0x10	Next Pattern	Select next pattern
Fast Forward	0x34		
-- (or 11)	0x0a	Set fixed colour	Set the fixed colour slot in fixed colour mode
0	0x00	Clear colour	Set the current max colour setting to zero (black)

1	0x01	Increase red component	Increase the red component of the max colour setting
2	0x02	Increase green component	Increase the green component of the max colour setting
3	0x03	Increase blue component	Increase the blue component of the max colour setting
4	0x04	Dim	Halve the red,green,blue components of the max colour setting
5	0x05	Random	Pick a random selection of red/green/blue components of the max colour setting
6	0x06	Brighten	Double the red,green,blue components of the max colour setting
7	0x07	Decrease red component	Decrease the red component of the max colour setting
8	0x08	Decrease green component	Decrease the green component of the max colour setting
9	0x09	Decrease blue component	Decrease the blue component of the max colour setting

Note: You can use either the Vol-/Vol+ or Rewind/FF button combination in selecting patterns. This helps in some remote controls with non-standard layouts or when you've configured a 'VCR' type remote to avoid conflicting with existing equipment. See section on 'Determining Remote Compatibility'.

### ***CMD\_SETPARM***

This sets the maximum colour RGB components of the maximum colour setting. Patterns will select random colours from zero to this RGB setting.

### ***CMD\_SETLED***

This command controls an individual PWM output to the PWM value specified. See chip pinout diagram for which LED number corresponds to which pin.

### ***CMD\_MOVELED***

This command moves an individual PWM output to the PWM value specified at the current pattern speed setting. See chip pinout diagram for which LED number corresponds to which pin.

### ***CMD\_SETPRESET***

In fixed colour mode, the Chan+/Chan- buttons are used to select from 16 different preset colours. The '-/--' or '11' key is used on the remote control to set

12 of these colours. This command allows you to do the same thing via program control, and to set the 4 reserved presets.

***CMD\_SETANNUN***

This command allows you to set/reset annunciators as well as set or clear different modes in the device. The data byte should have bit flags set as shown:

7	6	5	4	3	2	1	0
1- Reset Pinout to default	-	-	-	-	-	0 – Show on/off pattern 1- Don't down on/off pattern	0 – Blink group blue on pattern change 1- No blink on pattern change 0

Some notes on the CMD\_SETANNUN command.

- You can reset the unit to default state of showing the on/off pattern and blinking when changing patterns by sending CMD\_SETANNUN 0x00. This is useful if you have mis-imprinted your remote. You must have the on/off pattern displayed on power-on to clear an imprint.
- You might wish to turn off the on/off pattern or blinking for public installations where these could be distracting.
- If you don't show the on/off pattern, you cannot clear any imprinting or Annunciator settings.
- The default pinout is shown on the chip pinout diagram. You would clear to default by sending CMD\_SETANNUN 0x80

***CMD\_SETPINOUT***

This command allows you to set the grouping of the pins to the five RGB channels. For example in your printed circuit design it might be difficult to use the default setting without needing lots of vias or links, and you may wish to use this setting to make your design easier.

Each pin on the chip has a number from 0..15 (pin 5 is unused). You send the remaining pins in the order:

Group 1 Red, Group 1 Green, Group 1 Blue.....Group 5 Green, Group 5 Blue

If pins are repeated unpredictable results may occur.

See the chip pinout diagram for the default pins, their position an default channel grouping.

## Programmatic Control Examples

To do the following example tasks, send the command string as shown, including checksum. Refer to the above tables to see why.

Set all colours to a nice yellow:

0x53 , 0xff, 0xe0, 0x00, 0xXX

Set groups to Red, Orange, Greeny Blue, Blue, Magenta

0x2f, 0xff, 0x00, 0x00, 0xe0, x080, 0x10, 0xe0, 0x10, 0x00, 0x00, 0xff, 0xc0, 0x00, 0xf0, 0xXX

Turn on the on/off pattern and blinking between patterns Annunciator

0xe1, 0x00, 0xXX

Change to the fastest speed

0x71, 0x00, 0xXX

## Suggestions on Coding an Algorithm to transmit the Manchester Code

The IR decoding on the RGBSPCHIP5 chip uses the first two bits on the transmission to synchronize decoding of the rest of the byte. When using the programmatic interface this allows some leeway in the speed that data is transmitted. However if the bits in a byte are not sent at the same rate as the first two bits, there will be a data error as the checksum is unlikely to be read correctly.

Therefore it is important that all bits are transmitted with the same interval between them.

It may be tempting at first to program transmitter code thusly:

```
For Each Byte in Command String
    Get DataByte from Command String
    SendBit(1)
    SendBit(1)
    For Each Bit in DataByte
        SendBit(Bit)
    Next Bit
    WaitByte Interval
```

Next Byte

This is unlikely to work however, the reason being that the time used to interpret the commands two SendBit(1) used to do the synchronization will not be the same time to calculate the loop used to parse the data byte.

It is better to construct a 10 bit data byte (or 14 if you are simulating a real remote control) by setting bits 8 and 9 then transmit the whole 10 bits using the same loop as below:

Define DataByte as 16bit Value

```
For Each Byte in Command String
    Get DataByte from Command string
    DataByte = DataByte | 0x300
    For the 10 bits in DataByte
        SendBit(Bit)
```

Next  
Next

As example in the SourceBoost 'C' Compiler from PicAnt is shown below. The full code is shown in the appendices and is available on the RGBSuset Productions website:

```

unsigned char nCharDelay = 10; //for a 8Mhz 16F88, experiment for others.
//FUNCTION to send a data byte 'ch' out the IR transmitter to the RGBSPCHIP5 reciever
//USES:
//      set_bit () :- function to set a bit in a value
//      off_ir()  :- function to turn off IR transmitter
//      on_ir()   :- function to turn on IR transmitter
//      delay_10us() :- function to wait in 10uS increments
//      nCharDelay :- global defined value as delay time per bit
//RETURNS:
//      nothing
void tx_char(unsigned int ch)
{
    //Note 16bit data value used, so we can set bits 8,9
    unsigned char n;

    //set first two start bits for synchronization
    set_bit (ch,8);
    set_bit (ch,9);

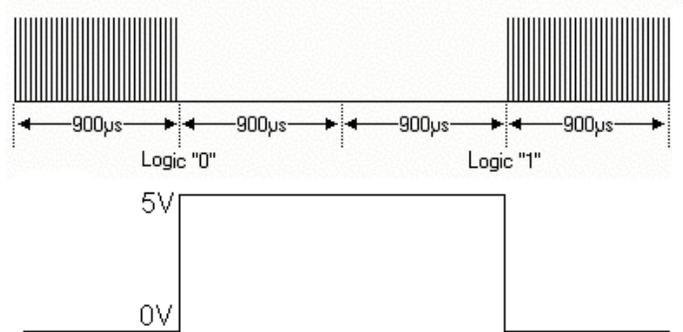
    n = 9;
    while(n<10)
    {
        //Do a test on bit 9 by ANDing with 0x200.
        //Same test each cycle through loop keeps the timing consistant for all 10 //bits.
        if (ch & 0x200) {
            //a '1' is sent as a 0->1 transition in Manchester code
            off_ir(); //turn off IR transmitter
            delay_10us(nCharDelay); //wait the bit time
            on_ir(); //turn on IR transmitter
            delay_10us(nCharDelay);
        }
        else
        {
            // a '0' is sent as a 1->0 transition
            on_ir();
            delay_10us(nCharDelay);
            off_ir();
            delay_10us(nCharDelay);
        }
        ch = ch << 1; //shift the bits along
        n--;
    }
    off_ir();
    delay_10us(nCharDelay); //wait a time before sending the next byte
    delay_10us(nCharDelay);
}

```

### Infra Red Receiver Circuitry

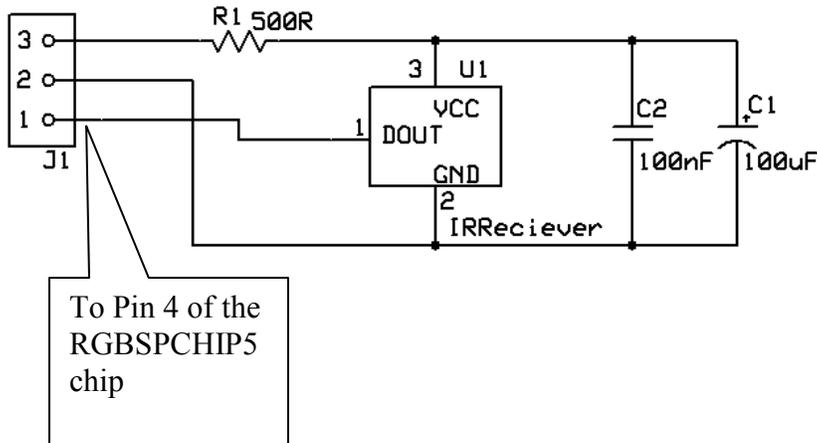
The carrier signal is usually best handled by commonly available IR modules especially designed to respond to a narrow range of infra-red frequencies and output a continuous high/low signal. They usually provided an inverted logic to the actual carrier being transmitted. The RGBSPCHIP5 is designed to connect to one of these. The receiver should output a 0V signal when a carrier is detected and a 5V signal when a carrier is not detected.

Eg:



Signal at RGBSP-CHIP5 rx pin, output of receiver module

The IR module or receiver circuit should connect to the rx pin of the RGBSPCHIP5 (pin 4). It should also have decoupling from the power supply as shown below, many IR receiver modules are very sensitive to transient variations on the supply rail.



If you have a pre-made assembly or kit, this circuit and connector should already be supplied.

### Determining Remote Compatibility

The unit will respond to RC5 or RC6 type remote controls, however most users probably will not know nor care what standard their remote conforms to. If you have a programmable

remote control, consult the manual that came with it on how to program your remote. Usually a sequence of buttons must be pressed to get the remote into program mode, then you would enter a code from a code book supplied with it for different brands of equipment.

- It is suggested that you write down the codes for the 'Philips' brand for TV/VCR type equipment. For each code, try the on/off button until the unit responds with the on/off pattern. Note that the on/off pattern can be disabled by the digital interface protocol, in which case the unit will simply switch on or off if the remote is recognized.
- If you find a code, keep going and note all codes that the unit responds to.
- Note that if you pick a 'VCR' type remote, you may find that the Vol-/Vol+ buttons are disabled. If this is the case, and you cannot choose a TV type remote because of compatibility problems with existing equipment you can also use the Rewind/Fast Forward buttons.
- Next check that code you have chosen does not interfere with any other equipment in the room.
- Finally check all the functions detailed in the 'Controlling the unit via a remote control' section work as expected. Some programmable remotes may not work as expected.

If you have several RGBSP units you may wish to see the section 'Imprinting your remote' on how to make each unit respond to only one code.

If you have the problem where your existing Audio/Visual equipment remote interferes with RGBSP unit, you may also like to see the section in 'Imprinting your remote'.

### Imprinting your remote

By default the RGBSPCHIP5 chip in your circuit or RGBSPVista5 board will respond to any RC5 compatible remote control.

However you may have existing RC5 equipment in range of where you are using your RGBSP product. Changing the colour or patterns on your lighting appliance may well change the volume or channel or your TV.....which would probably be rather annoying.

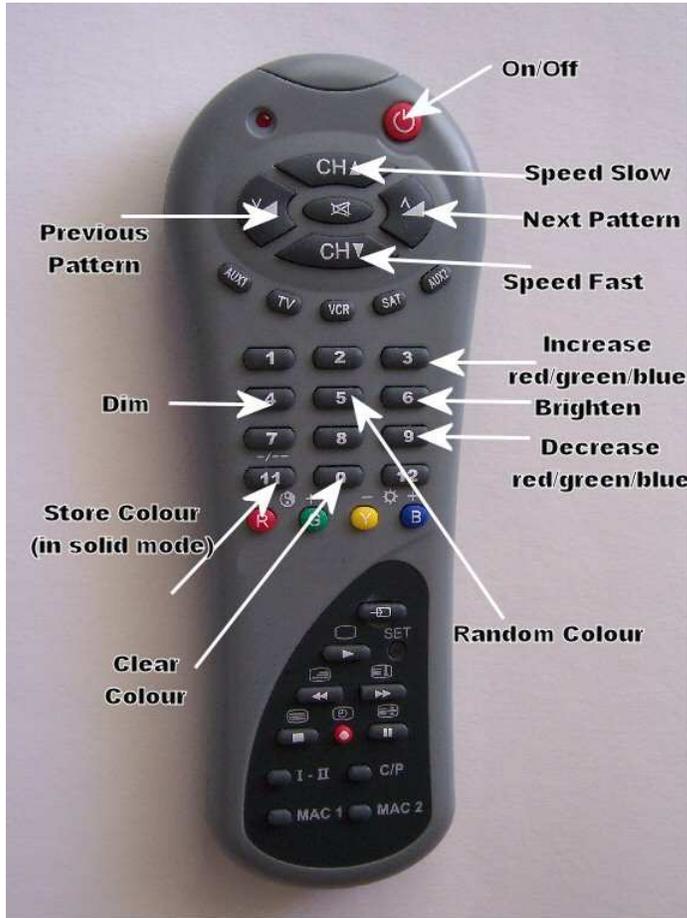
Alternately you may have several devices with RGBSPCHIP5 chips in them and want to control them separately with different functions on your remote control.

If this is the case you can use Imprinting to cause your device to respond to only one remote type. RC5 compatible remotes are classified into different devices defined by the 'address' field in the RC5 protocol, which allows a remote destined for a TV for example not to interfere with a VCR.

- When you switch on the device *from power*, and you press Chan- while the power up pattern is being displayed, the RGBSPCHIP5 will only respond to that remote control from now on. It is 'imprinted' like the myth a baby chicken is imprinted on the first thing it sees after hatching. It will stay in this state until the imprinting is cleared
- When you switch on the device *from power*, and you press Vol+ or FF button while the power up pattern is being displayed, the RGBSPCHIP5 imprinting is cleared. Other settings like pattern speed, max colour, current pattern, Annunciator setting are cleared to default as well

### IR Remote Keys and Operation

The image below shows a typical remote control and the buttons that function with the RGBSP unit:



The buttons have the following functions:

Button	Function
On/ Standby	<i>Toggles on/standby mode.</i> In standby, the unit will only respond to another press of this button and will ignore all others.
Vol+ Fast Forward	<i>Select next pattern</i> When the last pattern is already selected, then the first pattern is picked next. The unit will announce the change in pattern mode by blinking blue group 1 for pattern 1, group 2 for pattern 2 etc
Vol- Rewind	<i>Select previous pattern</i> As for Vol+, except if the first pattern is already selected then the last pattern is picked.
Chan+	<i>Slow pattern speed/Select next fixed.</i> There are 8 speeds. When changed group 1 will blink for a period, shorter for faster speeds, longer for the slowest speeds. In fixed pattern mode however this button will select the next of 16 fixed colours, which can be programmed using the ‘Store current RGB maximum’ key. When the last fixed colour is selected the unit will stay on that colour.

Chan-	<i>Increase pattern speed. Select previous fixed colour.</i> When the first fixed colour is selected, the unit will stay on that colour.
1,7	<i>Increase red/ decrease red pattern maximum</i> The maximum brightness of any pattern can be set in its component colours.
2,8	<i>Increase green/ decrease green pattern maximum</i>
3,9	<i>Increase blue/ decrease blue pattern maximum</i>
5	<i>Random RGB maximum selection.</i> A random maximum red, green and blue is selected.
4	<i>Decrease RGB maximum brightness.</i> The current settings for red, green and blue maximum is halved.
6	<i>Increase RGB maximum brightness.</i> The current settings for red, green and blue maximum is doubled.
0	<i>Clear RGB maximum.</i> The maximum goes to zero, and the display will go off. It will continue to respond to all keys however.
--/- or 11 key	<i>Store current RGB maximum</i> In fixed pattern mode, the current maximum setting is stored in EEPROM.

If pattern maximum keys are held down then maximum for each component will increase by a jump. Press the keys individually for fine control of the red, green and blue components. No other keys respond to being held down in any special way. Also note that you can use the Fast Forward/Rewind keys instead of Vol+/Vol- keys to set the pattern. Some remote controls disable the Vol+/Vol- buttons when they are configured as a 'VCR' style remote.

## On/Standby Pattern

When the unit is turned on, either from power or via the on/standby toggle, the unit will show the on pattern. The on pattern consists of the middle group brightening first, then slowly each group going outwards is brightened to full power. The unit then goes into the previously set speed, pattern and maximum colour settings, fading into the pattern if applicable.

During the on pattern after power is applied certain actions can be taken using the remote. Care should be taken not to use the remote during the on-pattern after turning on the power unless you wish to perform these actions. The actions are:

- *Reset, clearing all settings:* Hold the Chan- button to reset the units maximum colour to half (0x80) red, green and blue, Still Pattern, Speed 0, no imprinting, all annunciators on.
- *Remote Imprint:* Hold the Chan+ button down during on-pattern. See the section on 'Imprinting your remote' before using this function.
- *Annunciator Off:* The blinking of the pattern and speed annunciators can be turned off for ascetic reasons if desired by holding down the Vol- or Rewind button during on-pattern.
- *Annunciator On:* The annunciators can be turned back on during on-pattern by holding down the Vol+ or Fast Forward key.

When the unit is set to standby, first the middle group blinks red. Next the outer groups darken, then the inner groups darken. The inner group grows to maximum brightness, then

fades back to nothing. The unit will now only respond to another press of the on/standby button.

*(warning: The on/off pattern can be disabled or enabled via programmatic control, but not by a remote. The above functions can only be activated during the on pattern. If you turn off the on/off pattern then you will need to use the programmatic interface to re-enable it to clear imprinting/ Annunciator settings to default)*

### **RGB Maximum setting**

The RGBSP product range uses arrays of red, green and blue LEDs and control the brightness of these LEDs to simulate colours similar to the way the pixels on a television or computer monitor show colour. The RGB Maximum setting is the maximum brightness that LEDs in each colour will go to in any pattern.

This is visible in the range of colours that are displayed in dynamic patterns or the fixed colour shown in fixed colour mode. By changing this setting using the 1...9 keys, each pattern can have many variations. For example you may want a lava lamp that is only blue, or you would like warm colours like red/yellow/green in a bar that fades slowly. Perhaps you would like a simple magenta fixed colour.

Use the increase/decrease buttons for red, green and blue to mix. At first it is recommended that users select fixed pattern mode and select their RGB maximum until they are familiar with the concept of mixing red, green and blue to select a colour.

### ***Creating colours***

For example to create a yellow bar.

- Select fixed pattern mode by pressing the Vol+/Vol- until the last pattern is annunciated with a blue blink.
- Set the RGB maximum to zero by pressing the zero '0' button.
- Increase the red component to maximum by holding down the '1' key until it no longer changes the brightness.
- Increase the green component by holding down the '2' key for a short time, then stepping up or down using the '2' and '8' key by pressing and releasing these buttons until a good yellow found. A nice pleasing yellow will have mostly red, and only some green. Increasing to maximum green will give a sickly shade of yellow.

If you don't know what colour you want.....try pressing the '5' key and let the RGBSP unit pick one for you

Another example, to create a blue lava lamp

- Select lava pattern by using the Vol+/Vol- buttons. The first group should blink blue when lava pattern is selected
- Clear RGB maximum by pressing the zero key
- Increase blue to max by holding down the '3' key

Note how we were able to change the RGB Maximum setting in a dynamic pattern mode.

To have each group show random colours blinking slowly

- Select the random colour mode by using the Vol+/Vol- buttons. Normally this pattern runs quite fast and may appear sparkling on the faster pattern speed setting.
- Use the Chan+ button to slow down the pattern.

## Creating a white colour

This requires a section all by itself because as any photographer may tell you white is very subjective and dependant on your surroundings.

You may have seen ‘white balance’ setting on your video or still camera allowing you to select ‘Incandescent’, ‘Sunlight’ or ‘Fluorescent’ settings. This is because the human eye and mind tends to assume that its ambient environment is still sunlight....and adjusts its response. Of course if the ambient light is very saturated in one colour or another, then the mind won’t be fooled. However light sources that are not terribly biased to towards one particular hue can fool the mind into thinking it *is* white.

These settings allow the objective computers in your camera equipment to compensate the same way a human mind does, so that when you look at your photos, the pictures look ‘right’ when you print them out or look at them on the computer screen.

Although light sources we regard as white, like incandescent globes or fluorescent lights or in particular the sun are not saturated in any one hue, they can be identified using what is called colour temperature.

Low colour temperature white lights have a warmer, yellower colour than high colour temperature white lights like fluorescent lights which are often quite harsh and cold....of course we are not talking about real temperature here, as you can probably tell by the irony of having low temperature lights feeling perceptually warm!

The RGBSPVista5 unit with all LEDs set to maximum is bit of a bluish ‘white’. RGBSP-CHIP users will have a result that is dependant on their choice of LEDs and how bright they are. Some users may wish to pick very ‘warm’ greens and blues or select RGB LEDs that are balanced to a ‘warm white’.

For the RGBSPVista 5 it is a cold, high colour temperature though, allowing you to choose a bright spooky blue for special occasions. If you want a warmer white, the solution is simply to reduce the amount of blue in the mix. As you do so, you’ll notice the green starting to affect the ‘white’, and you’ll need to reduce the amount of green as well. For a warm white you’ll need mostly red, about half green and only a little bit of blue. This is easier to see when start from zero (ie press the ‘0’ button) on the remote. First increase the red to maximum, then green to a nice yellow, and then adjust the blue. You may want to re-adjust the green again and play around till you get a ‘white’ that suits your ambient environment.

## Index

### Communications

bit .....	1, 5, 6, 11, 12, 13, 19
command .....	5, 6, 7, 8, 9, 10, 11, 12
transmission.....	6, 12
Infra-Red	
protocol	
bit .....	1, 5, 6, 11, 12, 13, 19
IR .....	4, 9, 12, 13, 14, 16
transmission.....	5
RC5.....	4, 5, 6, 9, 14, 15
Remote.....	5, 7, 11, 12, 15, 18
TV .....	6, 15
VCR.....	5, 10, 15, 17
pattern .....	9

### Pattern

maximum .....	10, 16, 17, 18, 19
off pattern.....	11, 12, 15, 18
on pattern .....	11, 17, 18
speed .....	7, 8, 9, 10, 12, 15, 16, 17, 18
white .....	19
Power	
AC.....	3
halogen .....	3
transformer .....	3
Products	
assembly .....	1, 3, 4, 14
RGBSPCHIP5 .....	1, 4, 6, 7, 8, 12, 13, 14, 15
RGBSPVista5 .....	1, 3, 6, 7, 8, 15, 19