

```

;-----
; Title:                A12 - Color Rotation
; Description:          12 Chan RGB With Color Rotation
; Implementation:       RGB Mux 12 Using 15 Pins
; Micro-controller:     PIC16F628A
; Last Update:          10/17/2014
; Revision:             24
;
;
;
;

```



```

;
; A4 needs a pull up 10k resistor
; A5 (Master Clear) needs a pullup 10k resistor
; A4, A6, A7 sink to ground R,G,B thru 3 NPN, While B0-B7
; and A0-A3 source Anodes of RGB directly from PIC.
;-----

```

```

list                p=16F628A
#include <p16F628A.inc>

__CONFIG_CP_OFF & _WDT_OFF & _PWRTE_ON & _INTOSC_OSC_NOCLKOUT & _LVP_OFF
    errorlevel      -302                ; supress banksel warning messages during assembly
    errorlevel      -311                ; supress HIGH operator warning messages during assembly
    errorlevel      -305
    radix    dec                        ;if numbers are not identified default to decimal system

cblock0x20
    FrameCount
    LoopCount
    Speed
    Color1
    Color2
    Color3
    Col1
    Col2
    Col3
    C1
    C2
    C3
endc

#define bank0 bcf      STATUS,RP0
#define bank1 bsf      STATUS,RP0
RESET_VECTOR org      0x000

```

```

;-----
;                               Macros                               |
;-----
Paint      macro  Col1,Col2,Col3
            movlw  Col1<<4      ;user provided colors
            movwf  Color1       ;are read and placed
            movlw  Col2<<4      ;in left 4 bits of each
            movwf  Color2       ;color variable
            movlw  Col3<<4      ;at later time added to
            movwf  Color3       ;PORTA portion of data line
            call   Animate      ;main program
            call   Animate      ;is called 3 times
            call   Animate      ;to loop the same colors 3 times
            endm               ;before user provides 3 new colors

;-----
Pattern1    macro  L1           ;place pattern in data line L1
            movf   C1,W         ;place Color1 (A7-A4) in W
            addlw  L1/256       ;add color1 to left 4 bits of data
            movwf  PORTA        ;then place in PORTA to display
            movlw  L1%256       ;grab right 8 bits of data line
            movwf  PORTB        ;place them in PORTB to display
            endm

;-----
Pattern2    macro  L2           ;same as above for Color2
            movf   C2,W
            addlw  L2/256
            movwf  PORTA
            movlw  L2%256
            movwf  PORTB
            endm

;-----
Pattern3    macro  L3           ;same as above for Color3
            movf   C3,W
            addlw  L3/256
            movwf  PORTA
            movlw  L3%256
            movwf  PORTB
            endm

;-----
;                               Main Code Starts Here                               |
;-----
START
            movlw  b'00000111'   ;Set all ports as outputs
            movwf  CMCON         ;Disable comparators
            bank1
            clrf   TRISA         ;clear registers
            clrf   TRISB
            bank0

;-----
            movlw  d'10'         ;Change speed of rotation 1-255
            movwf  Speed         ;1=fastest, 255=almost stationary

Sequence
            call   MegaColor     ;Loop back to MegaColor
            goto   Sequence      ;and repeat color choices

```

```

;-----
;                               Color Entries                               |
;-----

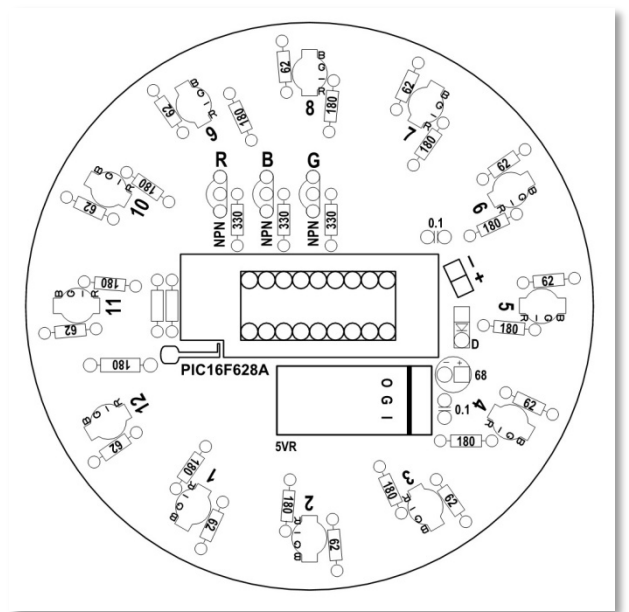
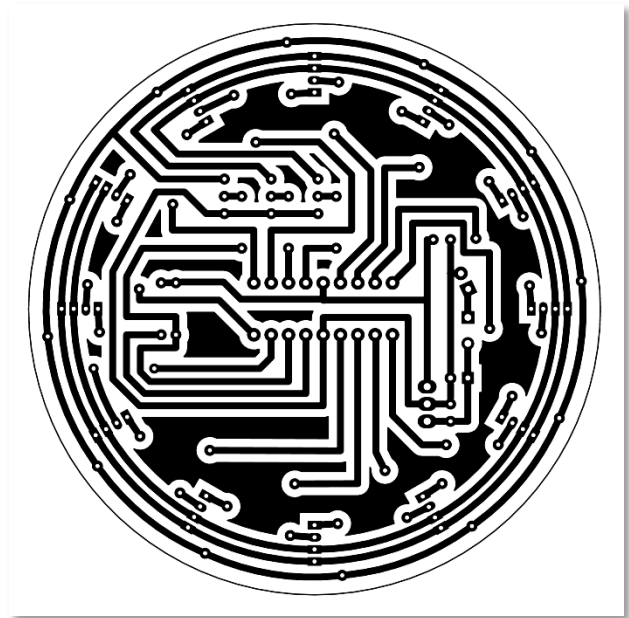
```

# MegaColor

```

Paint  b'0001',b'0100',b'1000' ;User provides 3 colors to rotate 3 times
Paint  b'0001',b'0101',b'0100' ;then select 3 new colors and repeat
Paint  b'0001',b'0001',b'0101' ;comment all the lines except one
Paint  b'0001',b'0100',b'0100' ;to loop the same colors over
Paint  b'0001',b'1000',b'0001' ;color choices are:
Paint  b'0001',b'0001',b'1000' ;0001=Red, 0101=Yellow, 0100=Green
Paint  b'0001',b'1001',b'1000' ;1100=Aqua, 1000=Blue, 1001=Magenta
Paint  b'0001',b'1001',b'1001' ;1101=White, 0000=Black
Paint  b'0001',b'0001',b'1001'
Paint  b'0101',b'0101',b'0001'
Paint  b'0101',b'0101',b'0100'
Paint  b'0101',b'0101',b'1100'
Paint  b'0101',b'0101',b'1000'
Paint  b'0101',b'0101',b'1001'
Paint  b'0101',b'1001',b'1100'
Paint  b'0101',b'0100',b'1100'
Paint  b'0101',b'1001',b'1001'
Paint  b'0101',b'1101',b'0001'
Paint  b'0101',b'1101',b'0100'
Paint  b'0101',b'1101',b'1000'
Paint  b'0100',b'1101',b'1000'
Paint  b'0100',b'1000',b'1000'
Paint  b'0100',b'0100',b'1000'
Paint  b'0100',b'1100',b'1000'
Paint  b'0100',b'1100',b'1100'
Paint  b'0100',b'1101',b'1100'
Paint  b'0100',b'1001',b'1001'
Paint  b'0100',b'0100',b'0101'
Paint  b'0100',b'0100',b'1100'
Paint  b'0100',b'0100',b'1001'
Paint  b'1000',b'1001',b'0001'
Paint  b'1000',b'1001',b'1001'
Paint  b'1000',b'1101',b'0001'
Paint  b'1000',b'1100',b'1100'
Paint  b'1000',b'1101',b'1001'
Paint  b'1001',b'1001',b'1101'
Paint  b'1001',b'0101',b'1101'
Paint  b'1001',b'1001',b'0100'
Paint  b'1001',b'1001',b'1100'
Paint  b'1001',b'1101',b'1100'
Paint  b'0001',b'0000',b'0000'
Paint  b'0101',b'0000',b'0000'
Paint  b'0100',b'0000',b'0000'
Paint  b'1100',b'0000',b'0000'
Paint  b'1000',b'0000',b'0000'
Paint  b'1001',b'0000',b'0000'
Return

```





```

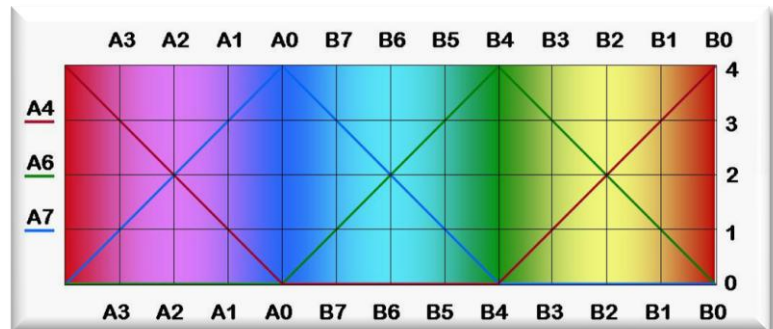
Frame1
    movlw    d'255'
    movwf    FrameCount
FC1      movfw    Speed           ;speed of rotation
    movwf    LoopCount
LC1      Pattern1  b'111000001111' ;Triangular shapes
    Pattern1  b'110000000111' ;are generated here
    Pattern1  b'100000000011' ;then rotated
    Pattern1  b'000000000001' ;in the next Frames
;
    Pattern2  b'000011111110' ;User can change
    Pattern2  b'000001111100' ;signal shape
    Pattern2  b'000000111000' ;as desired
    Pattern2  b'000000010000'
;
    Pattern3  b'111111100000' ;additional lines
    Pattern3  b'011111100000' ;can be added to
    Pattern3  b'001111100000' ;increase resolution
    Pattern3  b'000100000000' ;as needed
;
    decfsz   LoopCount,F        ;repeat loop to
    goto     LC1                ;slow things down
    decfsz   FrameCount,F
    goto     FC1
    return
;-----

```

```

Frame2
    movlw    d'255'
    movwf    FrameCount
FC2      movfw    Speed
    movwf    LoopCount
LC2      Pattern1  b'110000011111'
    Pattern1  b'100000001111'
    Pattern1  b'000000000111'
    Pattern1  b'000000000010'
;
    Pattern2  b'0001111111100'
    Pattern2  b'0000111111000'
    Pattern2  b'0000011110000'
    Pattern2  b'000000100000'
;
    Pattern3  b'1111111000001'
    Pattern3  b'1111111000000'
    Pattern3  b'0111111000000'
    Pattern3  b'001000000000'
;
    decfsz   LoopCount,F
    goto     LC2
    decfsz   FrameCount,F
    goto     FC2
    return

```



```

Frame3
    movlw    d'255'
    movwf    FrameCount
FC3
    movfw    Speed
    movwf    LoopCount
LC3
    Pattern1 b'100000111111'
    Pattern1 b'000000011111'
    Pattern1 b'000000001110'
    Pattern1 b'000000000100'
;
    Pattern2 b'001111111000'
    Pattern2 b'0001111110000'
    Pattern2 b'0000111100000'
    Pattern2 b'0000010000000'
;
    Pattern3 b'111110000011'
    Pattern3 b'1111100000001'
    Pattern3 b'1111000000000'
    Pattern3 b'0100000000000'
;
    decfsz   LoopCount,F
    goto     LC3
    decfsz   FrameCount,F
    goto     FC3
    return
;-----
Frame4
    movlw    d'255'
    movwf    FrameCount
FC4
    movfw    Speed
    movwf    LoopCount
LC4
    Pattern1 b'000001111111'
    Pattern1 b'000000111110'
    Pattern1 b'000000011100'
    Pattern1 b'000000001000'
;
    Pattern2 b'0111111110000'
    Pattern2 b'00111111100000'
    Pattern2 b'00011110000000'
    Pattern2 b'0000100000000'
;
    Pattern3 b'1111100000111'
    Pattern3 b'1111000000011'
    Pattern3 b'1100000000001'
    Pattern3 b'1000000000000'
;
    decfsz   LoopCount,F
    goto     LC4
    decfsz   FrameCount,F
    goto     FC4
    return
end

```