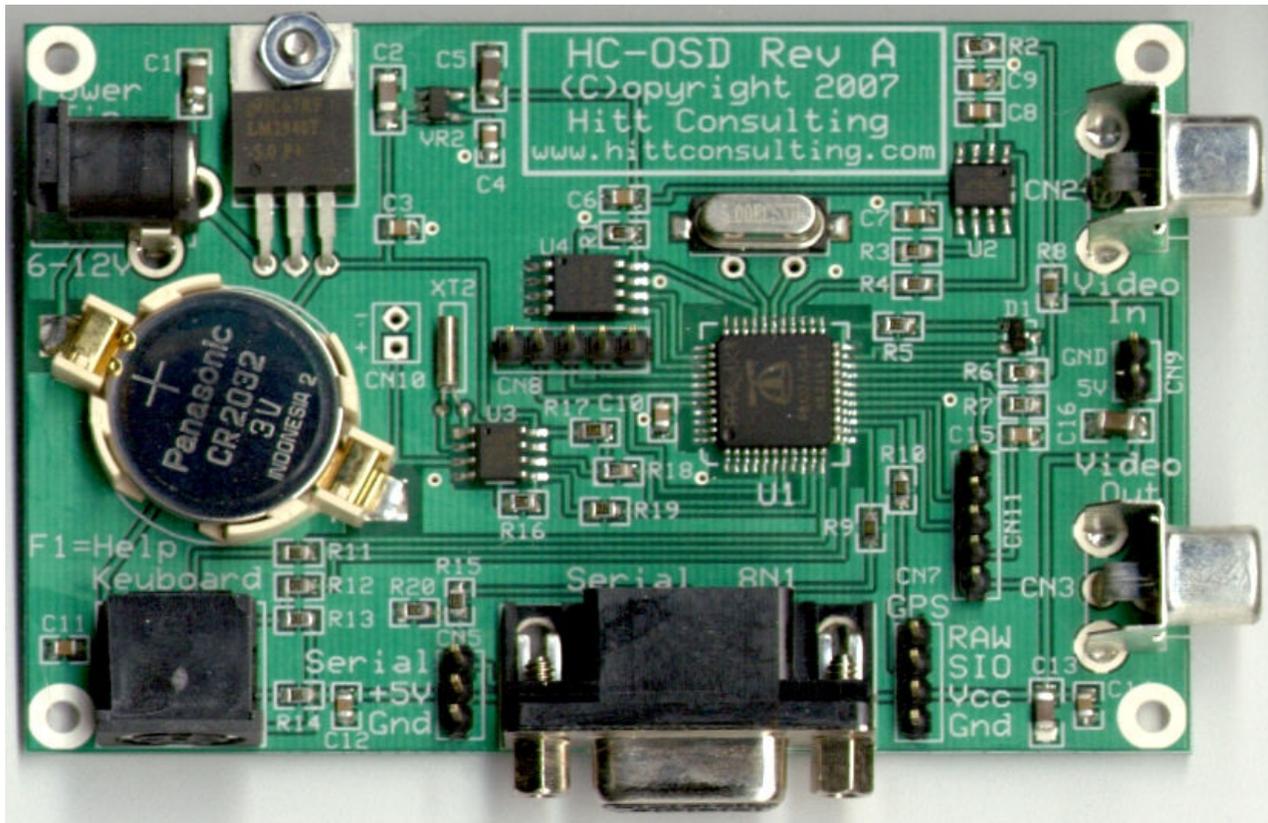


HC-OSD Video OSD Module

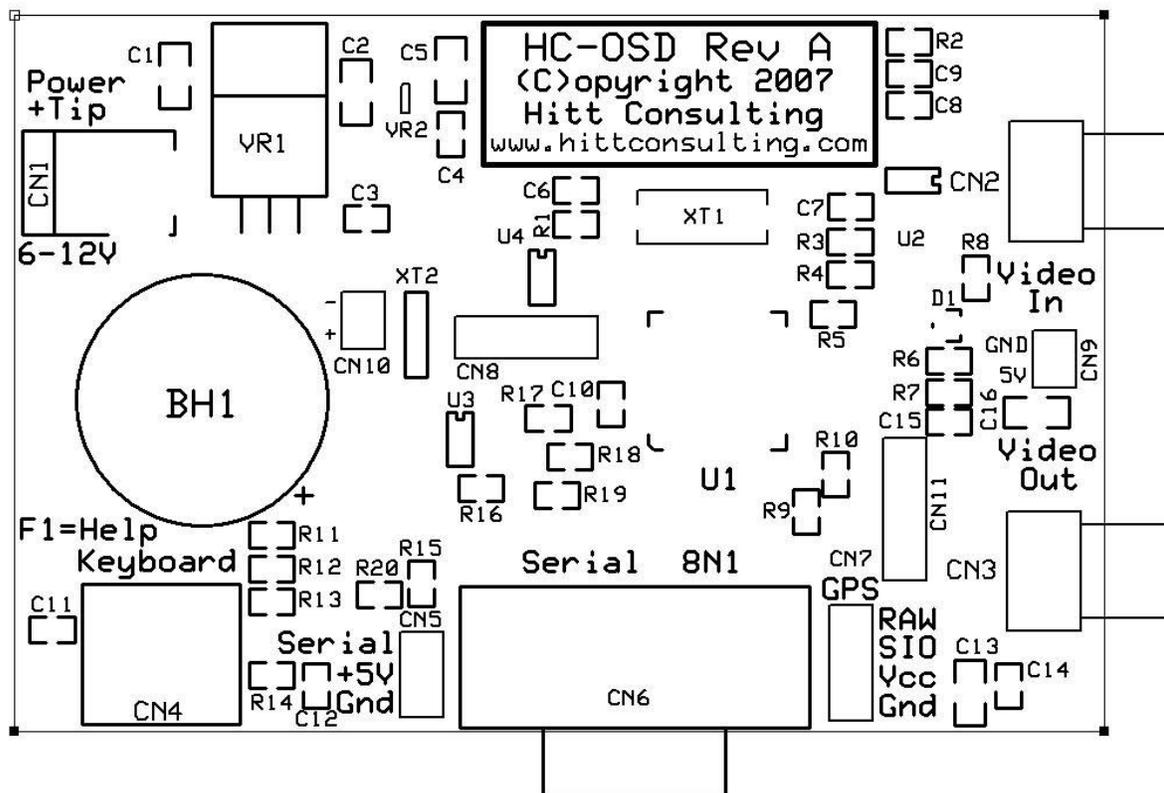


Firmware Ver 1.2 Datasheet Rev A

- * Video Overlay
- * PS/2 Keyboard Input
- * GPS Serial Input
- * TTL & RS232 Serial Input
- * Onboard Real-Time clock
- * High Resolution 256x192 pixels
- * Save/Load screen from EEPROM

Hitt Consulting

105 S Locust Street
Shiremanstown Pa 17011
info@hittconsulting.com
www.hittconsulting.com



SPECS

Power: 6.0 to 9.0VDC 50mA
 Size: 3.8" x 2.5" (not including connectors)

Controller: Parallax Propeller microcontroller operating at 80 MHz.

Display: 256(H) x 192(V) pixels
 Compatible with NTSC (USA) or PAL (Europe) video signals.

Graphics: 256x192 pixels. Each pixel can be transparent, black, white or gray(in overlay mode). Black and white in generate mode(no video input).

CONNECTOR FUNCTIONS

- CN1** Power Jack (2.1mm +Tip) 6 to 9 volts DC
- CN2** White RCA jack. This is the video input from a camera or other video source.
- CN3** Yellow RCA jack. This is the video output from the module. Connect this to the video input of your video display using an RCA cable.
- CN4** PS/2 Keyboard. NEVER connect or disconnect the keyboard when the module is powered.
- CN5** TTL Serial. Provides connection to a TTL Serial device (0V-5V). Serial data must be inverted at 19,200 baud, 8 data bits, 1 stop bit, no parity. Note that the module may be power by the +5V connection. Or the module will supply +5V if CN1 is used for power.

CN6 DB9 RS-232 Serial Data. Connection to a RS-232 serial port (such as a PC). Data format is 19,200 baud, 8 data bits, 1 stop bit, no parity. Note that the module does not send any data. It is a receive only device.

CN7 GPS input. This input is specifically for the Parallax GPS receiver, but should work for any GPS receiver that sends data at 4800 baud. This connector will supply power to the Parallax GPS also.

CN8 Propeller programming header. This header provides connection to the Parallax(r) Propeller(r) chip. Use this connection to reprogram the Propeller(r). Either the Parallax(r) USB2SER(Vss to right) or the PropPlug(Vss to left) may be used.

CN9 Camera Power. This header provides +5V and Ground for a camera that provides a video input.

CN10 Clock power. This header may be used to provide power to keep the clock functioning (if used for this purpose you MUST remove the 2032 coin cell battery). Or you may use the connector to supply 3V from the coin cell.

CN11 Expansion header. This header provides connection to +5V, +3.3V, Ground, and two propeller pins (P30 & P31). This is to be used for those who wish to customize the propeller firmware to provide additional capabilities.

OVERVIEW

The SX Video OSD Module is used to create text and high-resolution graphics “on-top-of” an input video signal. Popular uses include: time/date stamp, GPS coordinate tracking, security info, status displays, copyrighting video material and demonstrations.

The video module creates a display of 192 lines each consisting of 256 pixels. Each pixel can be one of: Transparent, White, Black, or Gray when a video input is present. When no video input is present the module will generate the video image and pixels will be either White or Black.

Thanks to the Parallax(r) Propeller the HC-OSD module is able to process multiple input. These are: PS/2 Keyboard, Real time clock, GPS input, TTL or RS-232 Serial input.

PS/2 KEYBOARD INPUT

With a PS/2 keyboard connect to CN4, characters typed on the keyboard will appear on the overlay screen. The keyboard is also used to configure the module using the function keys (F1 to F10).

F1 = HELP - Displays a help screen. Press the "Esc" key to clear the help screen. NOTE: The overlay will be clear after the help screen is displayed and cleared. Any information you have typed before pressing F1 will be gone.

F2 = RTC On/Off
F3 = GPS On/Off
F4 = Save
F5 = Load
F6 = Small Font
F7 = Large Font
F8 = Set Clock
F9 = PAL/NTSC
F10 = not used

REAL TIME CLOCK INPUT

The HC-OSD module contains a DS1307 real time clock with a 2032 lithium backup battery.

To set the real time clock, you must use the keyboard. Press the F8 key, the clock display in the upper left will show “--/--/-- --:--”. Using the number keys on the keyboard type in the month (use a zero first for months less than 10). Then type in the day of month (use a zero first for days less than 10). Then type in the last two digits of the year.

Pressing the F2 key on the keyboard will toggle the clock display on and off. This setting is saved to EEPROM so the next time the module is powered up the setting will be the same.

GPS SERIAL INPUT

When a GPS serial text output is connected to the CN7 connector (we recommend the Parallax® 28146 GPS receiver), the module will display the GPS information at the bottom of the screen. To toggle the display on and off press the F3 key on the keyboard. This setting is stored in EEPROM and is restored when the module is powered up.

SERIAL DATA INPUT

The serial data is received at a baud rate of 19,200 8 data bits, 1 stop bit, no parity. Character data is sent using standard ASCII codes. In other words on the Basic Stamp ® microcontrollers a simple message might be:

```
SEROUT SPin,Baud,["Hello"]
```

The TTL serial input (CN5) and the RS-232 serial input (CN6) cannot be connected at the same time.

The HC-OSD will buffer up to 256 characters. If data is sent non-stop while the HC-OSD is busy (for example saving or loading an image from EEPROM) then data may be lost.

COMMANDS

Received data values from 0 to 15 and 128 to 143 are commands. Some commands require additional data that is sent immediately following the command. For example the CrsrXY command must be followed by the character for the X value then the character for the Y value CHR(2)+CHR(X)+CHR(Y). Commands for values from 0 to 15 are the same as the DEBUG commands for the Basic Stamps ®, and the constants for DEBUG can be used.

The Null (128) value is useful to send at startup just in case a multi-byte command (like CrsrXY) was sent without the parameters. The next values sent would then be interpreted as the parameters for the CrsrXY command, instead of the commands intended. The Null value is simply ignored if received on it's own.

EXPANSION CONNECTOR

The CN11 connector provides for expansion for custom features. The connector has +5V, +3.3V, Ground and two propeller pins (P30 and P31).

The Propeller software is available for download from the website www.hittconsulting.com This software may be modified and used free for non-commercial uses only.

VIDEO INPUT AND OUTPUT

The video output of the module must be connected to a video input of another device. This is usually a yellow colored RCA jack on a TV, monitor, VCR or DVD player. Video inputs have an impedance of about 75 ohms, and this is needed to get the correct voltage from the module.

VIDEO OSD (OVERLAY) OR GENERATOR

The HC-OSD module will function as an On Screen-Display when there is a video signal present at the video in connector. When a video input is not recognized the module will switch to video generation mode. The module can be used as a video generator if needed.

NTSC/PAL VIDEO TIMING

The module is capable of generating either NTSC or PAL video timing. This setting is toggled by using the F9 key on the keyboard. When a video input is present, the module will auto-detect NTSC or PAL video and overlay with the correct timing. The F9 setting is only for when there is NOT a video input and the module is generating the complete video signal.

REGULATORY WARNINGS

This device is NOT FCC approved. It is not in finished product form. It is strictly intended for experimental purposes only. If you wish to use these modules in an actual product (a non-experimental capacity), the modules must first be designed into the product, then the whole product must be approved by the FCC.

LIMITED LIABILITY

Hitt Consulting's liability shall be limited to the purchase price of the HC-OSD module. The HC-OSD module may NOT be used in any situation where harm to humans, animals or property could result.

HEALTH WARNINGS

This product contains lead, a chemical known by the state of CA to cause cancer and birth defects and other reproductive harm.

CUSTOMIZATIONS

Customizations to the SX-Video OSD module are available from the developer Hitt Consulting. Please contact info@hittconsulting.com.

ERRATA

None

COMMANDS

Commands from 0 to 15 are the same as the Basic Stamp(R) DEBUG commands.
All parameters are sent as a single byte. CHR(X) or CHR\$(X).

| VALUE | NAME | PARAMETERS | DESCRIPTION |
|-------|------------|--------------------------------------|---|
| 0 | Cls | (none) | Clears the screen and moves cursor to 0,0 |
| 1 | Home | (none) | Moves cursor to 0,0 |
| 2 | CrsrcXY | (2) X, Y | Moves cursor to specified coordinates |
| 3 | CrsrcLf | (none) | Moves cursor 1 position to the left |
| 4 | CrsrcRt | (none) | Moves cursor 1 position to the right |
| 5 | CrsrcUp | (none) | Moves cursor 1 position upward |
| 6 | CrsrcDn | (none) | Moves cursor 1 position downward |
| 7 | Ignored | (none) | Ignored |
| 8 | Bksp | (none) | Moves cursor 1 position to the left (destructive) |
| 9 | Tab | (none) | Moves cursor to next tab stop (0,8,16) |
| 10 | LF | (none) | Linefeed; Moves cursor down 1 line |
| 11 | ClrEol | (none) | Clears from cursor to end-of-line |
| 12 | ClrDn | (none) | Clears from cursor to end-of-screen |
| 13 | CR | (none) | Carriage Return; Moves cursor to start of next line |
| 14 | CrsrcX | (1) X | Moves cursor to specified column |
| 15 | CrsrcY | (1) Y | Moves cursor to specified row |
| 128 | Null | (none) | Ignored |
| 129 | LineColor | (1) Color | Sets the drawing color(0-3) |
| 130 | LineWidth | (1) Width | Sets the drawing width (1-15) |
| 131 | MoveToXY | (2) X, Y | Sets the next drawing position to X(0-255), Y(0-191) |
| 132 | PlotXY | (2) X, Y | Plots a point at X(0-255), Y(0-191) |
| 133 | LineToXY | (2) X,Y | Plots a line from the current point to X(0-255), Y(0-191) |
| 134 | BarXY | (2) X,Y | Plots a bar from the current point to X(0-255), Y(0-191) |
| 135 | Arc6 | (6)Rx, Ry, Ang, AngStep, Steps, Mode | Plots an arc with the current point as the center. See text for detailed explanation. |
| 136 | TextNormal | (none) | Sets the text parameter to default values. |
| 137 | TextSize6 | (6)Sx, Sy, lx, ly, Fw, Bw | Sets the text parameters. See text for detailed explanation. |
| 138 | TextColors | (2) ForeColor, BackColor | Sets the foreground and background colors for text. |

| VALUE | NAME | PARAMETERS | DESCRIPTION |
|---------|----------------|---------------------|--|
| 139 | SaveToEEPROM | (1) Position | Saves the screen overlay image to EEPROM. See text for detailed explanation. |
| 140 | LoadFromEEPROM | (1) Position | Loads a screen overlay image from EEPROM. See text for detailed explanation. |
| 141 | BitMapXY | (varies) X, Y, Data | Receives bitmap data. See text for detail explanation. |
| 142 | RTCOn | (none) | Enables the time and date display from the real time clock. |
| 143 | RTCOff | (none) | Disables the time and date display from the real time clock. |
| 144 | GPSON | (none) | Enables the GPS coordinates display. Only valid if module is receiving GPS data. |
| 145 | GPSOff | (none) | Disables the GPS coordinates display. |
| 146 | NTSCMode | (none) | Enable NTSC timing during generate mode. |
| 147 | PALMode | (none) | Enable PAL timing during generate mode. |
| 148-191 | Reserved | ??? | Reserved for future firmware updates. Do not use if customizing firmware. |
| 192-255 | Custom | ??? | Uses these values if you customize the firmware. Future firmware updates will not use these. |

DETAILED COMMAND USAGE

X, Y: These are overlay coordinates. X can range from 0 (left of screen) to 255 (right of screen). And Y can range from 0 (top of screen) to 191 (bottom of screen).

SetColor(129): Valid colors are 0 to 3. 0=Transparent during overlay, black during generation; 1=Black; 2=White; 3=Gray during overlay, white during generation.

Arc6(134): Rx = Radius in the X (horizontal) direction; Ry = Radius in the Y (vertical) direction; Ang = Starting Angle (in brads 1/256 of a circle); AngStep = Angle step size (in brads); Steps = Number of steps to take in arc; Mode: 0=Plot points; 1=Lines to points; 2=Lines between points; 3=Lines from points to center; Normally Mode=2 is used to draw arcs.

TextSize6(136): Sx=Scale in X (horizontal) direction; Sy=Scale in Y (vertical) direction; lx=Increment in X (horizontal) direction; ly=Increment in Y (vertical) direction; Fw=Foreground Width; Bw=Background Width

SaveToEEPROM(137):

LoadFromEEPROM(138): EEPROM storage has 9 positions (0 to 8) This allows you to store 9 overlay images. Position 0 is loaded automatically when the module is powered up.

BitMapXY(139): This command is used to send a bitmap image to the overlay. The bitmap starts at the current position. The X parameter is the number of bytes across (1-64 there are 4 pixels per byte), and the Y parameter is the number of pixels down (1-191). After the X and Y parameters you must send X*Y bytes of data that represents the image to be displayed. Each byte represents 4 pixels. Data is arranged from left to right, top to bottom.

Basic Stamp ® Example Programs

```

=====
' File..... HC-OSD_Test
' Purpose... To test the HC-OSD serial input
' Author.... Terry Hitt
' E-mail.... terry@hittconsulting.com
' Started... Mar 26, 2007
=====
' NOTE: The HC-OSD has a 256 byte serial buffer. Do not send more than
' 256 bytes of data without allowing the HC-OSD time to process the data.
=====
' Stamp to HC-OSD connections
'
' BS2      HC-OSD
' ---      -
' P15 ---- Serial (CN5)
' GND ---- Gnd   (CN5)
'
'{$STAMP BS2}
'{$PBASIC 2.5}

#SELECT $STAMP
#CASE BS2, BS2E, BS2PE
  T1200    CON    813
  T2400    CON    396
  T4800    CON    188
  T9600    CON     84
  T19K2    CON     32
  T38K4    CON     6
#CASE BS2SX, BS2P
  T1200    CON   2063
  T2400    CON   1021
  T4800    CON    500
  T9600    CON    240
  T19K2    CON    110
  T38K4    CON    45
#CASE BS2PX
  T1200    CON   3313
  T2400    CON   1646
  T4800    CON    813
  T9600    CON    396
  T19K2    CON    188
  T38K4    CON    84
#ENDSELECT

Transparent  CON    0
White        CON    1
Black        CON    2
Gray         CON    3

SevenBit     CON   $2000
Inverted     CON   $4000
Open         CON   $8000

' Command      Value  Parameter Bytes  Notes
'-----
Null           CON    128 '      0      No effect
LineColor      CON    129 '      1      Color (0 = Transparent/Black; 1=White; 2=Black;
3=Gray/White)
LineWidth      CON    130 '      1      Width (1-15)
MoveToXY       CON    131 '      2      XCoord, YCoord
PlotXY         CON    132 '      2      XCoord, YCoord
LineToXY       CON    133 '      2      XCoord, YCoord
BarXY          CON    134 '      2      XCoord, YCoord
Arc6           CON    135 '      6      XRadius, YRadius, Angle, AngleStep, Steps,
ArcMode(0=Points; 1=Line to points; 2=Lines between points; 3=Lines center to points)
TextNormal     CON    136 '      0      Normal font
TextSize6      CON    137 '      6      XScale, YScale, XStep, YStep, ForegroundWidth,
BackgroundWidth
TextColors     CON    138 '      2      Foreground Color, Background Color
SaveToEEPROM   CON    139 '      1      Location 0-8 (0=StartUp)
LoadFromEEPROM CON    140 '      1      Location 0-8 (0=StartUp)
BitMap         CON    141 '      2+Data  Bit mapped data
RTCon          CON    142 '      0      Turns Real-Time-Clock display On
RTCOff         CON    143 '      0      Turns Real-Time-Clock display Off
GPSOn          CON    144 '      0      Turns GPS display On (must receive GPS data)
GPSOff         CON    145 '      0      Turns GPS display Off
NTSC           CON    146 '      0      Sets NTSC Generate Mode

```

```

PAL          CON      147 '      0          Sets PAL Generate Mode

Baud        CON      T19K2 + Inverted
OSDPin     PIN      15

I VAR Word

Start:
HIGH OSDPin
PAUSE 3000
SEROUT OSDPin, Baud, [Null, Null, Null, Null, RTCOFF, GPSoFF]
DO
  SEROUT OSDPin, Baud, [CLS]
  ' Setup to draw medium size text
  SEROUT OSDPin, Baud, [TextSize6, 2, 2, 16, 20, 2, 3]
  SEROUT OSDPin, Baud, [MoveToXY, 0, 0, "Hitt Consulting"]
  SEROUT OSDPin, Baud, [MoveToXY, 25, 25, "Presents..."]
  ' Setup to draw large size text
  SEROUT OSDPin, Baud, [TextSize6, 5, 5, 42, 60, 5, 7]
  SEROUT OSDPin, Baud, [MoveToXY, 0, 50, "HC-OSD"]
  ' Return to normal size text (small)
  SEROUT OSDPin, Baud, [TextNormal]
  SEROUT OSDPin, Baud, [MoveToXY, 0, 125, "* Video Overlay (NTSC/PAL)"]
  SEROUT OSDPin, Baud, [MoveToXY, 0, 140, "* PS/2 Keyboard Input"]
  SEROUT OSDPin, Baud, [MoveToXY, 0, 155, "* Real Time Clock"]
  SEROUT OSDPin, Baud, [MoveToXY, 0, 170, "* GPS Input"]
  ' Draw a small checkerboard pattern as a bitmap
  SEROUT OSDPin, Baud, [MoveToXY, 200, 180, BitMap, 2, 8]
  SEROUT OSDPin, Baud, [64+4, 64+4, 16+1, 16+1, 64+4, 64+4, 16+1, 16+1, 64+4, 64+4, 16+1, 16+1, 64+4,
64+4, 16+1, 16+1]
  PAUSE 5000
  SEROUT OSDPin, Baud, [CLS]

  ' Draw a moire pattern using LineToXY
  FOR i = 0 TO 191 STEP 2
    SEROUT OSDPin, Baud, [LineColor, Transparent, MoveToXY, 0, 0, LineToXY, 255, i]
    SEROUT OSDPin, Baud, [LineColor, White, MoveToXY, 0, 0, LineToXY, 255, i+1]
  NEXT
  FOR i = 255 TO 0 STEP 2
    SEROUT OSDPin, Baud, [LineColor, Transparent, MoveToXY, 0, 0, LineToXY, i, 191]
    SEROUT OSDPin, Baud, [LineColor, White, MoveToXY, 0, 0, LineToXY, i-1, 191]
  NEXT
  PAUSE 5000
  SEROUT OSDPin, Baud, [CLS]

  ' Draw a spiral using ARC6
  SEROUT OSDPin, Baud, [LineColor, White]
  FOR i = 0 TO 192 STEP 3
    SEROUT OSDPin, Baud, [MoveToXY, 127, 95, Arc6, I, I, I, 32, 8, 2]
  NEXT
  PAUSE 5000

LOOP

```