SPECTACULAR

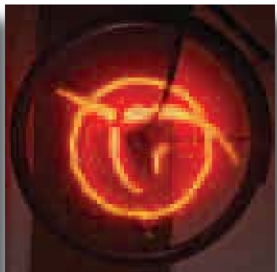
- Support for in-circuit serial programming (ICSP)
- 32 LEDs on each side of each PC board (64 LEDs per board, 192 LEDs total)
- Displays a one-kilobyte image (32 LEDs x 256 radial 'raster lines')
- All LEDs can be driven with 20mA at 100% duty indefinitely. This produces a very bright image.
- Firmware shuts the circuit down automatically when the voltage gets too low, to prevent damage to rechargeable battery packs
- The PC boards fit 26-inch bike wheels or larger.

BIKE WHEEL POV DISPLAY

This project uses POV to produce a spectacular glowing display on a rotating pushbike wheel as you ride along. So what is POV? It stands for 'persistence of vision'. It's a term that's applied to devices that rely on the human eye's tendency to 'see' an image for a short time after it has disappeared.

Designed by Ian Paterson





HOW WOULD YOU LIKE to own the most talked-about pushbike in the school/street/suburb. . . galaxy? Build this POV display and you'll be well on the way.

You really have to see it to believe it – and we've even made it easy for you. As well as the images printed here, there are several more you can view online at www.ianpaterson.org/projects

OK, you've now seen them and you'd have to agree that they look pretty spectacular. You want to do the same for your bike? Just make sure you keep it chained up, because everyone will want it!

Persistence of vision

You probably don't realise it, but you use POV every day – when you watch TV. Movies also take advantage of this phenomenon.

The TV and movie picture is not continuous, – rather (in the case of TV), 25 individual pictures are displayed every second. But your eyes and brain cannot follow the 25 individual frames of picture per second – instead, they 'fill in the gaps' and you 'see' full motion, non-jerky video.

If you slowed down those frames to, say, 10 per second, then you would be able to see the period between each frame and it would become jerky – just like the old-time movies where the hero moves like a Thunderbirds puppet.

Let's take this one step further. Say you had a moving light – we'll make it an LED because they can be turned on and off very quickly – which you flashed on, very briefly, once per second. You'd see this as flashes of light moving along a path. If you changed that to 10 flashes per second, you'd probably still see flashes, but very much closer together. Make that 50 flashes per second and the flashes would all flow into one another. You'd see it as a continuous line of light – even though your brain knows full well that it is flashes you are viewing.

That's *persistence of vision,* and this is the basic theory behind this project. Rows of LEDs are made to flash too quickly for your brain to process, so they appear to be permanently on. The rows of LEDs are mounted on PC boards fixed to a bicycle wheel, so they follow a circular path as the wheel rotates.

By using some clever circuitry to switch the LEDs on and off at particular moments, a pattern or picture can be created – in fact, the display is almost unlimited. It can be anything from geometric shapes to text, cartoon characters and even very high contrast pictures (see examples below).

In a nutshell

The display consists of three PC boards, each with a row of 32 LEDs on each side (a total of 64 LEDs). These boards are mounted radially in/on the spokes of a pushbike wheel and each has a battery pack mounted near the wheel's hub.

Talk about a WOW! factor: this three-high static display uses different coloured LEDs in each wheel to reveal three different patterns. The 'rider' powers the first wheel and the second and third wheels are driven by friction between the tyres.

A Hall effect sensor measures the rotational speed of the wheel by sensing a small magnet fixed to the bike frame. This sensor sends speed pulses to a microcontroller, which then turns the individual LEDs on and off in such a way that a static image appears to float inside the wheel.

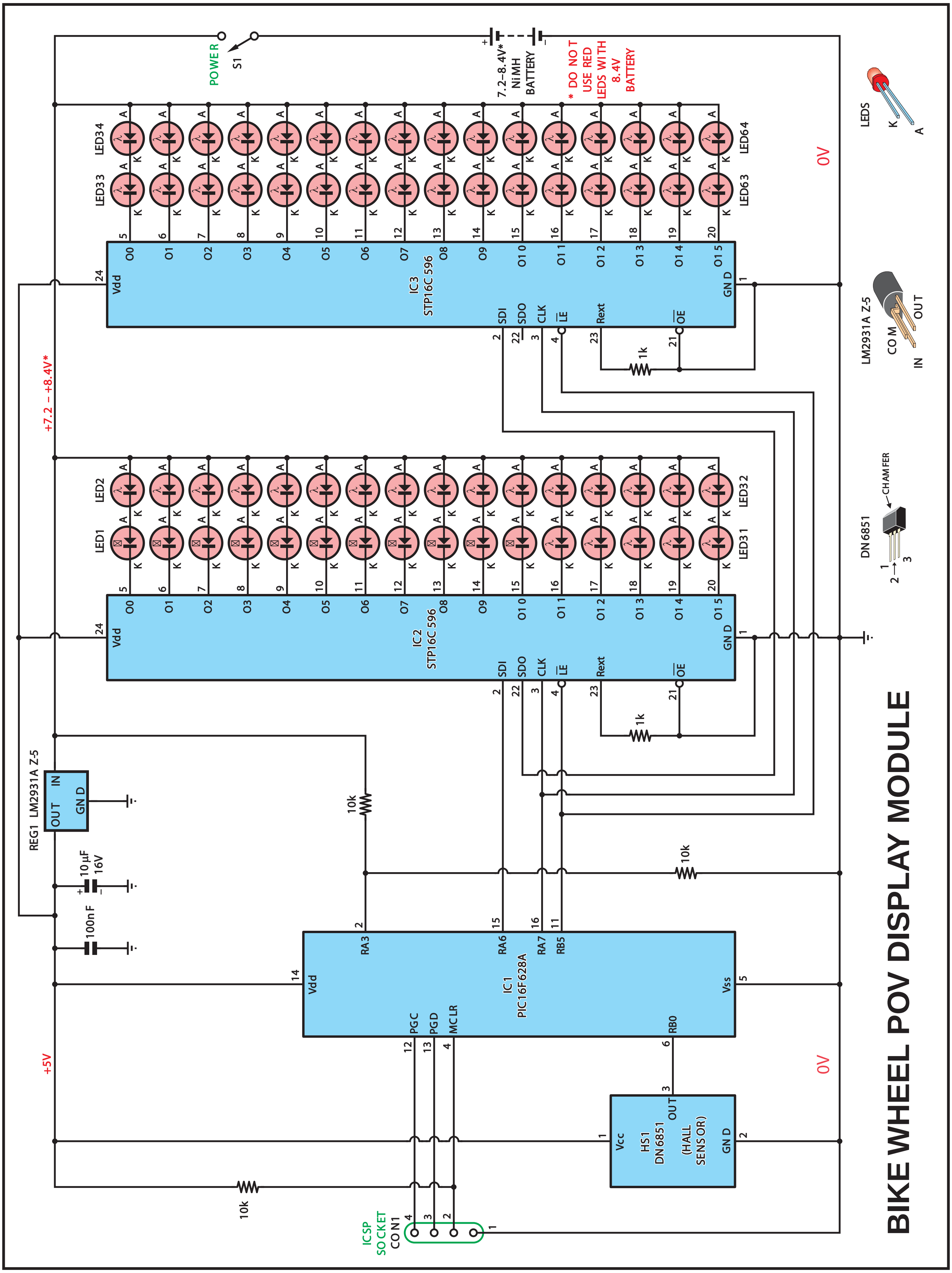
Circuit description

The complete circuit for one POV display module is shown in Fig.1. Three such modules are required, arranged so that





Here are just a few of the images generated by the author: (from left) pagan star, ET, invisible unicorn, Saturn and evolution!


BIKE WHEEL POV DISPLAY MODULE
POWER
S1
7.2–8.4V*
NiMH
BATTERY
* DO NOT USE RED LEDS WITH 8.4V BATTERY
+7.2 – +8.4V*
+5V
0V
LED1
LED2
LED31
LED32
LED33
LED34
LED63
LED64
IC2
STP16C 596
IC3
STP16C 596
Vdd
GND
SDI
SDO
CLK
LE
Rext
OE
O0
O1
O2
O3
O4
O5
O6
O7
O8
O9
O10
O11
O12
O13
O14
O15
1k
10k
REG1 LM2931A Z-5
IN
OUT
GND
10µF
16V
100nF
IC1
PIC16F628A
RA3
RA6
RA7
RB5
Vss
RB0
PGC
PGD
MCLR
HS1
DN6851
(HALL SENSOR)
Vcc
ICSP
SOCKET
CON1
LEDS
K
A
LM2931A Z-5
IN
COM
OUT
DN6851
CHAMFER

each is mounted 120 degrees from the other around the wheel, between the spokes. With the exception of the trigger magnet and battery pack, all components mount on these three PC boards.

The modules, or PC boards, each contain 64 high-brightness LEDs, 32 on each side. An LED on one side is connected in series with an LED on the other, so that the same image is seen on both sides of the bike.

In control

The LEDs are under the control of a PIC16F628A microcontroller (IC1). It is this microcontroller which not only stores the image to be displayed, but also outputs it to two STP16C596 shift registers (IC2 and IC3), which in turn drive the LEDs.

If each LED pair was driven with a dedicated output line, the microcontroller would have to have a very large number of output lines.

Hence, this circuit uses 16-bit constant-current LED sink drivers (IC2 and IC3) which can drive 16 outputs and allow multiple devices to be cascaded together. The STP16C596 also has a separate storage register that allows one set of data to be displayed while the next set is being loaded.

Four lines are used to control the LED outputs: serial data input (SDI), clock (CLK), latch enable ($\overline{\text{LE}}$) and serial data output (SDO).

Each pulse of the clock line causes the data to be 'shifted' over by one place and each pulse of the latch enable line causes the LED outputs to reflect the contents of the shift register.

One kilobyte of image data is stored in the program memory area of the microcontroller and is read by way of a look-up table. The firmware uses four interrupt routines:

- One to provide the time interval between radial raster lines
- One to increment a counter for timing the wheel rotation interval
- One to reset all counters and update the raster interval value every time the Hall effect sensor is triggered
- One that shuts down all LEDs when the battery voltage gets too low.

Fig.1 (left): one POV display module – three are required for the whole project. With 64 LEDs per module it looks daunting, but there are only 12 other components in each!

In fact, after the initial start-up routine, virtually every part of the firmware's execution runs inside an interrupt routine.

Hall effect

We haven't discussed the DN6851 Hall effect sensor yet. Its purpose is to measure the speed of the wheel and supply the appropriate timing pulses to IC1. It's triggered each time it passes a small magnet attached to the bike frame. Its output pulse is sensed by input RB0 on IC1

Timing values for the radial raster line interval are retrieved from a look-up table that exists in the microcontroller's program space. Data for the look-up table is generated with a QBasic program, although you only need to run this program if you want to experiment with different timing values.

When using a 7.2V battery pack, it's better to use a low dropout regulator, such as the National Semiconductor LM2931AZ-5, than the commonly used 78L05. It will continue to provide a solid 5V for the microcontroller even when the battery is at 6V. This is important, because if the supply voltage to the microcontroller drops, so does the internal reference voltage, which would prevent the voltage sensing routine from working properly.

A number of flow charts have been created to illustrate the logic in Spoke POV's various firmware routines, but since our space is limited, these can all be accessed on the website mentioned earlier.

QBasic programs

In addition to the microcontroller firmware, two Qbasic programs are required for setting the timing values and converting image data so that they can be incorporated in the firmware.

POVSLOPE.BAS creates the timebase look-up table. The table produced by this program is linear, so the only parameters one needs to be concerned with are slope and offset. Note that the timing data supplied in the sample firmware is reasonably accurate, so you should only use POVSLOPE.BAS if you plan to experiment with different timing values.

POVIMAGE.BAS is used to convert a raster image into radial data in the form of a series of 'RETLW B'xxxxxxxx';' commands that can be copied and pasted directly into the POV assembly code. The image data is read one pixel at a time as a series of 32 concentric rings. Each group of eight rings ends up occupying one memory page.

Because of the limitations of QBasic, it has been made to read headerless RAW files. The images must be 700×700 pixels, eight bits per pixel, with the pixels being either pure black (0×00) or pure white (0×ff).

Such a file can be created with Photoshop or many other graphics programs. When you've finished creating the image, the final file size should be exactly 490,000 bytes.

To stop the LEDs from lighting up when the bike is stationary, the last raster line is always set to zero (off).

Because the firmware stops incrementing the raster line counter when it reaches the last line in the image, having all LEDs off in that line will cause them to remain that way until the next trigger pulse from the Hall effect sensor.

Software

The software files will be available for download via the *EPE* Library site, access via **www.epemag.com** and also from the author, Ian Paterson – see Firmware panel. Pre-programmed PICs are available from Magenta Electronics – see their advert in this issue for contact details.

Construction

The double-sided printed circuit boards for the Bike Wheel POV Display are available as a set of three (code 711) from the *EPE PCB Service*. The component layout for one board (the other two are identical) is shown in Fig.2, together with top and underside photographs.

After checking the PC board for any copper track defects or solder 'bridges', start construction with the three 10kΩ and two 1kΩ resistors, followed by the 100nF and 10μF capacitors. Of these, only the 10μF radial electrolytic capacitor is polarised. Note, this capacitor is mounted on its side on the PCB, with its leads bent 90° to allow the leads to enter their respective holes in the board – see Fig.2 and photos. Since this is a double-sided PC board, we should mention that, apart from a row of LEDs, the components mount on the side with the writing in the copper.

Next, solder in the three IC sockets (the right way round) and two 'keyed' pin connectors, followed by the polarised regulator (REG1) and Hall effect sensor. One of the trickiest parts of this project is soldering the Hall effect sensors without damaging them.

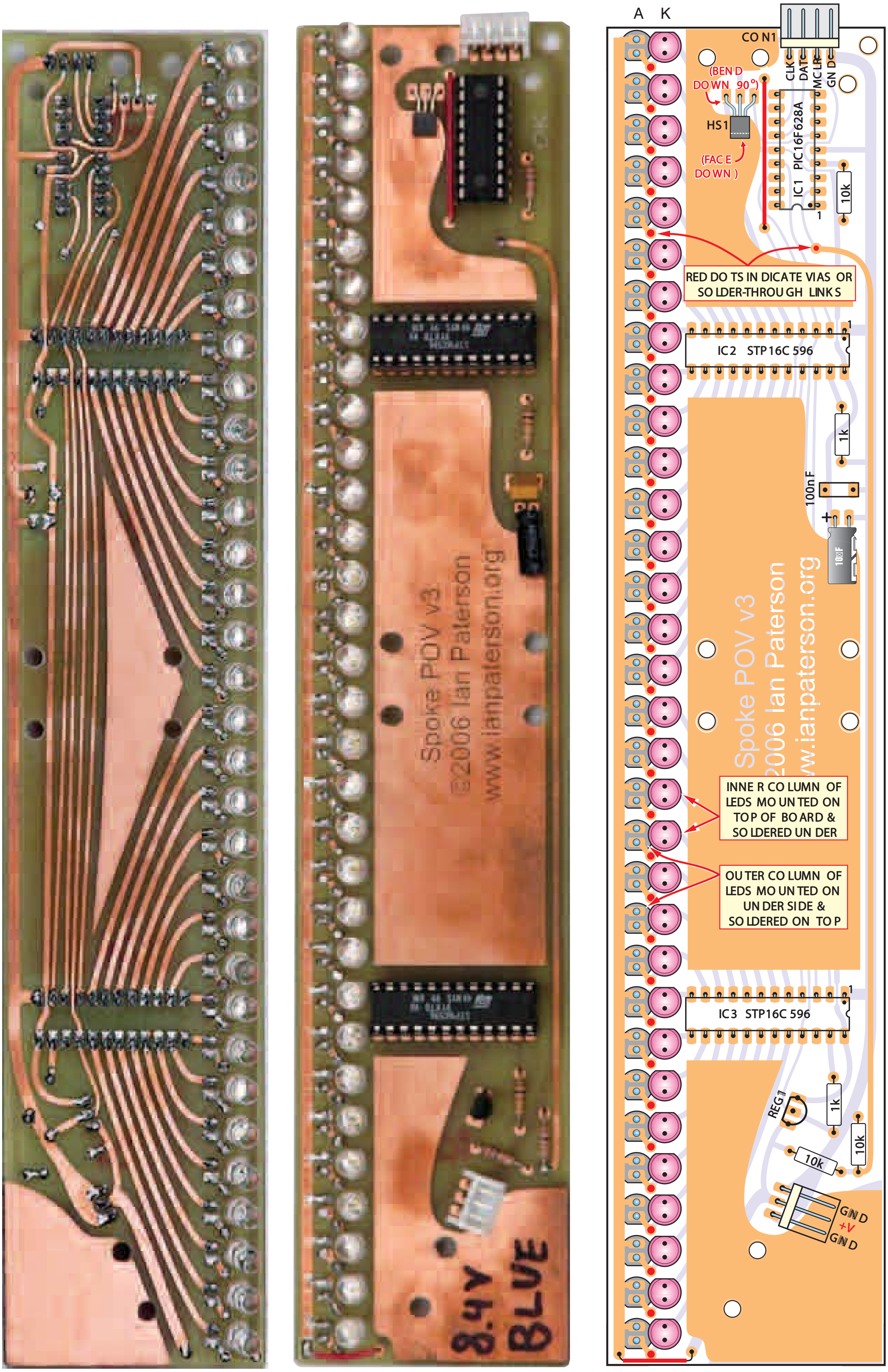


Fig.2: the PC board component overlay (shown from the component side) with matching top and bottom-side photographs of the PC board. The fingerprints are an optional extra! Seriously, the boards should be coated with a PC board (ie, solder-through) lacquer immediately they are made to prevent this from happening – especially as these boards will be out in the weather on the pushbike. In fact, we'd even go so far as to give the whole thing a good spray once finished – making sure you don't get lacquer in the two connectors.

FIRMWARE

Ian Paterson's firmware for this project – 628h.asm, povslope.bas and povimage.bas can be down-loaded from his website at www.ianpaterson.org/projects or from the EPE website, www.epemag.com

Because they are sensitive to both mechanical and thermal stress, you must use great care when attaching them to the circuit board. Their leads must be bent down 90° towards the face which has a chamfered edge on its top. This means that the face will actually be towards the PC board surface when fitted.

When bending the leads, you must hold the sensor lead with needle-nose pliers between the plastic case and the point at which the lead is being bent. This is to prevent mechanical stress at the point where the leads enter the sensor's case.

When soldering, you must also use needle-nose pliers as a heatsink to prevent damage from excessive heat. Once the sensors have been successfully soldered onto the board, there is little risk of further damage.

Soldering the LEDs

You have probably noticed that we have left the LEDs until last. That's because there are a lot of them and they can also be a bit tricky to solder. There are 32 LEDs to be soldered to *each side* of the PC board.

Note first, which lead is the anode and which is the cathode of the LED – there is a flat spot on the body of the LED next to the cathode (labelled 'K' on the circuit diagram). Also, the anode (A) lead is usually longer.

On the top (component side) of the PC board, the LEDs are arranged with their cathodes (K) oriented towards CON1 (the 4-pin connector) while on the bottom side, the reverse is true – see Fig.2.

The LEDs are controlled in pairs, one for each side of the board. This ensures that your chosen POV image can be viewed from both sides of the bike. The LED pairs are connected in series with small jumper wires (red dots on the component layout diagram) through the PC board that serve the same purpose as a PC board 'via' – they connect together the copper tracks on both sides of the PC board where required.

The biggest challenge in soldering these jumpers is that the heat from your soldering iron will travel along the wire and melt the connection on the other side of the board. I found it helpful to use those 'third hand' soldering aids with alligator clips to hold the wire in place.

If you are able to obtain or make PC boards with vias, then these jumpers are not necessary.

Finally, plug the three ICs into their sockets. Be careful to line up the notch in the end of the IC with

Pictured here are the same PC boards shown opposite; this time fixed to their backing 'plate', ready for mounting on the wheel. Note the semi-circle notches at the bottom end to fit into the axle. The top end is rounded to fit against the rim.

Parts List – POV Display *

3 PC boards, each 50 x 245mm, code 711, (available from the *EPE PCB Service* as a set)
3 18-pin IC sockets
6 24-pin IC sockets
3 7.2V or 8.4V 700mAh (or higher) battery packs (do not use 8.4V with red LEDs) – see text
3 4-pin PCB keyed header pin strips (CON1)
3 3-pin PCB keyed header pin strips
3 miniature On/Off slider switches for battery packs
3 magnets – see text
Material for backing plates – see text

Semiconductors
3 PIC16F628A microcontroller programmed with 628h.hex (IC1)
6 STP16C596 LED driver (IC2, IC3) – see alternatives below
3 DN6851 Hall effect sensors (HS1) – see alternatives below
3 LM2931AZ-5 low-dropout regulators (REG1)
192 high brightness LEDs (LEDs 1 to 64)

Capacitors
3 10µF 16V radial electrolytics
3 100nF MKT polyester or monolithic

Resistors (5%, 0.25W carbon film)
9 10kΩ
6 1kΩ

Alternative Parts

STMicroelectronics STP16C596 LED driver alternatives:
Allegro A6276EA
Maxim MAX6969ANG
Maxim MAX6971ANG

Panasonic DN6851 Hall effect sensor alternatives:
Melexis US5881EUA
Allegro A1101LUA-T
Allegro A1103LUA-T

* This list is for all three modules

In daylight, you can see the arrangement of the PC boards and batteries inside the spokes of the wheel. The PC boards, mounted 120° apart around the wheel, fit against the axle and are secured at the rim end via a couple of cable ties onto the spokes. It's important to keep the battery packs (which ever form you use) close to the axle to prevent the wheel getting out of balance.

the notch in the end of the socket. A second check is a small paint dot or indent beside pin 1 – you must make sure this goes where pin 1 is shown on the component overlay.

Loading an image

Since this POV design stores the image in program memory space, the microcontroller must be re-programmed every time you want to load a new image. The process is as follows:

- Create a 700×700 pixel, eight-bits per pixel image and save it with an eight-character filename.
- Edit POVIMAGE.BAS so that it references the new image and run the program. It will save its output with a .ASM extension.
- Copy and paste the .ASM output into the POV firmware file (628h.asm).
- Compile it to produce a .HEX file and program the POV board via the four-pin in-circuit serial programming (ICSP) connector. This connector does not supply power to the board during programming, so you must supply power from a battery pack or an external supply.

Testing

Test the operation of the POV board before fixing it to the spokes. It's a lot easier to fix mistakes on the bench than on the bike! Of course, the microcontroller should be programmed at this stage.

Apply power and wave a magnet in front of the Hall effect sensor. You should see the LEDs illuminate. They won't make much sense (ie, there will be no picture to see) but at least you will know the microcontroller is doing its job.

If they don't light up, turn the magnet over and try again. The faster you wave the magnet in front of the sensor, the faster the LEDs should flash. If this test fails to illuminate the LEDs, the most likely causes are a defective Hall effect sensor or a bad program.

Batteries

The battery voltage needs to be high enough to allow the regulator to provide 5V for the microcontroller and also just high enough to allow the LED drivers to deliver up to 20mA through each LED pair. Try using a 7.2V battery pack for LEDs with a low forward voltage (such as red) and 8.4V for other colours (such as white and blue). Be sure not to use a battery voltage that's more than about 2V higher than twice the forward LED voltage, otherwise the LED drivers may be damaged.

In the prototype, battery packs were made up from AA NiMH cells. I used 700mAh cells, but with 2500mAh now available, 1000mAh and even 1500mAh are becoming quite cheap. The larger the capacity, the longer your display will last.

You can use six cells (for 7.2V) or, as long as you don't use red LEDs, seven cells (8.4V) in your battery packs – it's more a case of getting a suitable holder. All three packs should be the same weight to avoid unbalancing the wheel.

An alternative, albeit a bit heavier, is to buy 7.2V or 8.4V battery packs intended for radio controlled models. High power (3500mAh+) ones are expensive, but you can often find lower capacity types on eBay for less than £10. Just make sure you mount them so they can't fly off!

Wheel mounting

The accompanying photo shows the position of the PC boards on the bike wheel. It's important to note that the inner edge of the PC board sits right up on the axle and that the whole thing is centred between the spokes, so that the board is right in the centre of the wheel.

To mount the PC boards in the wheel, a protective backing was made out of 3mm sintra (often used as a rigid backing onto which printed material can be mounted), one side was covered with anti-static plastic (cut from a motherboard bag), and was attached to the solder side of the PC boards using plastic cable ties.

We are not sure if the anti-static plastic is of any real benefit, but it was used as a precaution in case a static charge builds up on the sintra as the wheel spins.

At one end of the sintra, a crescent-shaped notch was cut to match the radius of the wheel front hub shaft. On the other end, a notch for the spoke nipple was also cut.

All that is needed to secure a PC board to the wheel is two cable ties at the spoke nipple end – the other end stays put because the crescent-shaped notch engages around the wheel hub.

To keep the hub end of the boards in place, two short sections of plastic hose were used. These were slit down one side, wrapped around the hub shaft and attached with cable ties. These act as spacers that prevent the boards from sliding laterally along the length of the hub shaft.

Note: these boards will fit a 26-inch or larger wheel. Also, when using three boards, it's easier to mount them in a wheel with a number of spokes that's divisible by three (eg, 36 spokes).

Mounting the magnet

To trigger the Hall effect sensors, the author used a stack of four magnets from an old 3.5-inch hard drive.

The stack of magnets were placed on the inside of one of the bike forks, immediately above the region under which the Hall effect sensor passed, then secured in place with a strip of tape.

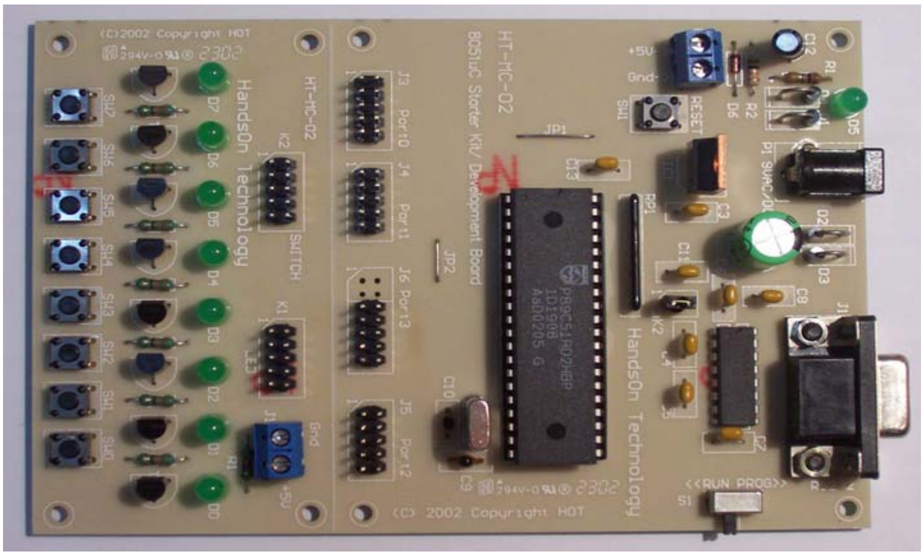
Other suitable magnets would be one or more of the rare-earth or so-called 'super magnets' which are enormously powerful for their size. ***EPE***

More information?

There are plenty more notes, flowcharts, firmware and graphics on the author's website, just set you browser to:
www.ianpaterson.org/projects

Low Cost 8051μC Starter Kit/ Development Board *HT-MC-02*

HT-MC-02 *is an ideal platform for small to medium scale embedded systems development and quick 8051 embedded design prototyping.* ***HT-MC-02*** *can be used as stand-alone 8051μC Flash programmer or as a development, prototyping and educational platform*



Main Features:

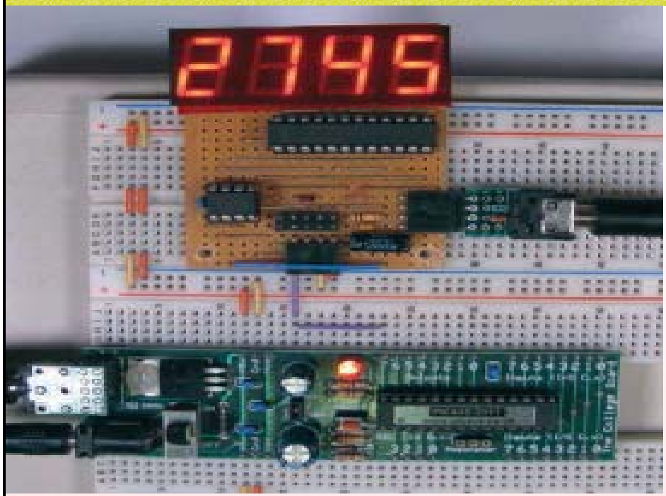
- 8051 Central Processing Unit.
- On-chip Flash Program Memory with In-System Programming (ISP) and In Application Programming (IAP) capability.
- Boot ROM contains low level Flash programming routines for downloading code via the RS232.
- Flash memory reliably stores program code even after 10,000 erase and program cycles.
- 10-year minimum data retention.
- Programmable security for the code in the Flash. The security feature protects against software piracy and prevents the contents of the Flash from being read.
- 4 level priority interrupt & 7 interrupt sources.
- 32 general purpose I/O pins connected to 10pins header connectors for easy I/O pins access.
- Full-duplex enhanced UART **–** Framing error detection Automatic address recognition.
- Programmable Counter Array (PCA) & Pulse Width Modulation (PWM).
- Three 16-bits timer/event counters.
- AC/DC (9~12V) power supply – easily available from wall socket power adapter.
- On board stabilized +5Vdc for other external interface circuit power supply.
- Included 8x LEDs and pushbuttons test board (free with ***HT-MC-02*** while stock last) for fast simple code testing.
- Industrial popular window ***Keil*** C compiler and assembler included (Eval. version).
- Free ***Flash Magic*** Windows software for easy program code down loading.

PLEASE READ ***HT-MC-02 GETTING STARTED MANUAL*** BEFORE OPERATE THIS BOARD
INSTALL ACROBAT READER (AcrobatReader705 Application) TO OPEN AND PRINT ALL DOCUMENTS

HandsOn Technology

http://www.handsontec.com

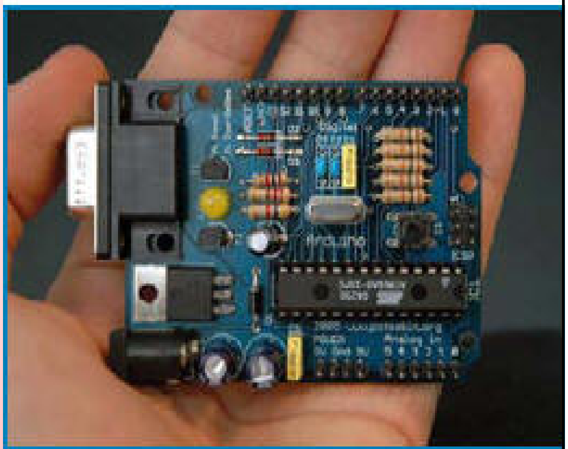
creativity for tomorrow's better living...



HandsOn Technology is a manufacturer of high quality educational and professional electronics kits and modules, uController development/evaluation boards. Inside you will find Electronic Kits and fully assembled and tested Modules for all skill levels. Please check back with us regularly as we will be adding many new kits and products to the site in the near future.
Do you want to stay up to date with electronics and computer technology? Always looking for useful hints, tips and interesting offers?

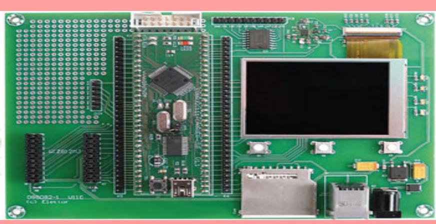
Inspiration and goals...

HandsOn Technology **provides a multimedia and interactive platform for everyone interested in electronics. From beginner to diehard, from student to lecturer... Information, education, inspiration and entertainment. Analog and digital; practical and theoretical; software and hardware...**
HandsOn Technology provides Designs, ideas and solutions for today's engineers and electronics hobbyists.



Creativity for tomorrow's better living...

HandsOn Technology believes everyone should have the tools, hardware, and resources to play with cool electronic gadgetry. HandsOn Technology's goal is to get our "hands On" current technology and information and pass it on to you! We set out to make finding the parts and information you need easier, more intuitive, and affordable so you can create your awesome projects. By getting technology in your hands, we think everyone is better off
We here at HandsOn like to think that we exist in the same group as our customers >> curious students, engineers, prototypers, and hobbyists who love to create and share. We are snowboarders and rock-climbers, painters and musicians, engineers and writers - but we all have one thing in common...we love electronics! We want to use electronics to make art projects, gadgets, and robots. We live, eat, and breathe this stuff!!
If you have more questions, go ahead and poke around the website, or send an email to sales@handsontec.com. And as always, feel free to let your geek shine - around here, we encourage it...


http://www.handsontec.com