



<http://www.instructables.com/ex/i/2806EBB22B071029BC6B001143E7E506/?ALLSTEPS>

The *01/Vatch*, because...

“there are 10 types of people in the world, those who read binary, and those who don't” – a slashdot tag line.

The *01/Vatch* is a binary wrist watch with an LED display. Additional features are accessible through a scrolling menu system on its 3x4 LED matrix. Current features include: voltage meter, binary counter, club mode and time display. The watch is fully programmable. Future firmware upgrades will include: stopwatch/timer, alarm, bicycle speedometer/odometer, data logging, and an advanced configuration menu.

All the project files are in the .zip archive on this page. Schematic and PCB in Cadsoft Eagle format. Firmware in mikroBasic. The text of this instructable is included as .odt (OO.org/open text) and .pdf files. The top-layer PCB art (mirrored) is included as a .PDF ready for toner transfer or foto process. It is copied several times on a single sheet because I have to double-up on transparencies.

The *01/Vatch* was inspired by the Mini Dotclock, and a subsequent conversation in the comments area: <http://www.instructables.com/ex/i/47F2F12223BA1029BC6B001143E7E506>

This is also a half step towards a surface mount nixie watch I am working on. The *01/Vatch* project is an introduction to surface mount components and time keeping logic without the added complexity of a

nixie tube power supply. (Read more here:

<http://www.instructables.com/ex/i/2C2A7DA625911029BC6B001143E7E506/?ALLSTEPS>)

A little googling turned up this binary watch at thinkgeek:

<http://www.thinkgeek.com/gadgets/watches/6a17/>

The *01/\atch* is based on a PIC16F913/6. This PIC was originally chosen because it had a hardware LCD driver. I thought that I could turn the LCD driver into a LED multiplexer with a few transistors. This turned out not to be the case. Its still a good choice because it has tons of programming space and very few limited I/O pins. The F913 is about \$2.00 at Mouser.

PIC16F913 Details:

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en020199

PIC16F916 Details (same as 913, with more program space):

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en020201

PIC16F913/6 Datasheet (PDF format):

<http://ww1.microchip.com/downloads/en/DeviceDoc/41250E.pdf>

Display

The binary display is made of 12 LEDs in a 3x4 matrix. Each column of four LEDs represents a four bit 'nibble', or half byte. Each column can display 0-15 in binary (1+2+4+8=15). Time is displayed in the three rows as hours/tens of minutes/minutes. This is not true binary, but a simplified subset that makes the watch easier to read. The thinkgeek watch, for example, uses 'truer' binary to represent minutes with a whole byte. Whichever I might prefer, the true geek would display time using the Unix epoch, in binary! [http://en.wikipedia.org/wiki/Unix_timestamp]

The LED multiplex is straightforward. Rows (4) connect to pins of the PIC through current limiting resistors. Only one current limiting resistor is used for each row because only one LED per row is ever lit. The LEDs are run at 20ma, using 56 ohm resistors (56ohm @ 3 volts=2ma). The LEDs could be run higher because they are multiplexed, the datasheet listed something around 40ma. I find them to be too bright at only 20ma-multiplexed.

Columns (3) are connected to ground by NPN transistors. The transistors are switched by PIC pins through 1Kohm resistors. The multiplex functions by grounding a column of LEDs through the transistor while lighting the correct LED rows for that column. This is repeated for each column in short succession, making the matrix appear to be continually lit. PIC Timer0 is drives the multiplex. It counts to 256 then changes row values and the grounded column.

Transistor:

NPN Transistor, NPN/ 32V/ 100mA, (Mouser #512-BCW60D \$0.05).

LED Selection

On this watch, yellow and red '1206' size LEDs were used with a 56 ohm current limiting resistor. The colors were chosen for low cost. Red, yellow, and orange LEDs are about 10 cents each, while blue LEDs are 40 cents and above. Besides, LED blue is decidedly uncool now. If you find some purple, let me know.

The picture shows the 5 LED types I auditioned.

Mouser Part #	Manufacturer	Color	Cost
859-LTST-C171KRKT	Lite-On SMT LED	Red, Clear	\$0.130
859-LTST-C171KSKT	Lite-On SMT LED	Yellow, Clear	\$0.130
859-LTST-C150KFKT	Lite-On SMT LED	Orange, Clear	\$0.130
638-121SURCS530A28	Everlight LED SMD	Red Water Clear	\$0.110
638-1121UYCS530A28	Everlight LED SMD	Yellow Water Clear	\$0.110

Everlight red and yellow were used on the prototype watch. I like Lite-On red and orange better, they will be used on the next watch I make.

Interface/Buttons

A geeky watch needs a geeky interface. Capacitive touch sensors are all the rage right now, but require quite a few extra components. Instead, I went with a Darlington transistor based touch sensor with pin headers as the contact point. What's geekier than a pin header? Nothing.

I first saw the idea here: [<http://www.kpsec.freeuk.com/trancirc.htm>]:

“A Darlington pair is sufficiently sensitive to respond to the small current passed by your skin and it can be used to make a touch-switch as shown in the diagram. For this circuit which just lights an LED the two transistors can be any general purpose low power transistors. The 100kohm resistor protects the transistors if the contacts are linked with a piece of wire.”

A PNP transistor was added to this simple design (in place of the LED in the diagram) so that it could give a high/low output to the PIC. A pull-down resistor was added between the PIC pin and ground to help prevent false button presses. This switch is solid state, water proof, and low power – with the added geekiness of pin headers.

Switches are de-bounced using Timer2 on the PIC. When a switch is pressed, Timer2 (8 bit timer) is started with a 16 prescaler and 16 postscaler. On Timer2 interrupt the PIC checks to see if the buttons are still pressed. After two consecutive interrupts with no buttons pressed the timer is stopped and the buttons are configured for further input.

The top switch is connected to the PIC interrupt pin. Input on this pin can bring the PIC out of sleep mode. This lets us use a neat power management technique: the PIC is in low power mode when the display is not in use. Input on the buttons wakes the PIC and resumes operation.

Transistors:

Darlington Transistor, SOT-23, (Mouser #512-MMBT6427, \$0.07).

PNP Transistor, SOT-23, (Mouser #512-BCW89, \$0.06).

Time keeping

Microchip app note 582 describes the basic principles behind a low power, PIC based clock.

[http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en011057]

The clock is simple and elegant. A 32.768kHz watch crystal is connected to the timer1 oscillator pins of the PIC. Timer1 is great for this because it can increment even while the PIC is sleeping. Timer1 is setup to count to 65536 (2 seconds at 32.768kHz) and wake the PIC from sleep with an interrupt. When the PIC wakes, it increments time by two seconds. The PIC is only active and consuming power for a brief time every few seconds.

I used a cheap quartz watch crystal from Citizen. I thought the Citizen name might give my watch legitimacy. The CFS206 (12.5pf) has about +/- 1.7 minutes accuracy per year (20ppm). Two 33pF capacitors complete the external crystal circuit. 33pF is probably a tad much, but it was available locally at a reasonable price. A better crystal could be used for more accurate time.

Crystal: Citizen KHz Range Crystals, 32.768 KHZ 12.5pF, (mouser #[695-CFS206-327KFB](#) , \$0.30).
Capacitors: 2x33pF, 1206 SMD.

Voltage Meter

As if we hadn't sunken to the depths of geekerie with a binary watch, we slap on a voltage reference and input pin to make a voltage meter.

The voltage reference is the Microchip MCP1525. This is a 2.5 volt reference with an operating range of 2.7 to 10+ volts. In the pictured watch the TO-92 package is used, though future watches will use the surface mount version (SOT-23). The reference is powered by a PIC pin so it can be turned off to conserve power.

At this point we can measure up to 2.5 volts using the PIC's Analog Digital Converter. We take this a step further and add a resistor voltage divider to the multimeter input. Using two resistors (100K/10K) we divide the input voltage by 11 giving a new input range of ~30 volts. This is a good point that encompasses all the low voltages we are likely to encounter (1.2/1.5 volt batteries, 3 volt coin cells, 5 volt logic, 9 volt batteries, and 12 volt power rails). A 22Kohm resistor could be substituted for the 10K resistor giving a smaller range but higher resolution. *The spreadsheet included with this instructable can help you choose resistor values.*

Ground and measurement probes connect to the programming header at the back of the watch.

MCP1525 Details:

http://www.microchip.com/stellent/idcplg?idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en019700

Programming Header/External Connections.

The watch is 'programmable'. An ICSP header is brought out to the back so that new firmware can be installed. The header is a row of low-profile female pin sockets I found at my local electronics store. The same thing can be had by cutting a quality DIP socket in half the long way. I connect my ICSP plug with a pin-header "gender-changer" - insert a piece of pin-header into the socket, then connect the ICSP plug to the pin header. *You will need an ICSP programmer to put new software in the watch. A*

simple JDM2 ICSP programmer is included with the Cadsoft Eagle files.

When not used for programming, the ICSP header can be used for data collection, event logging, etc. All of the ICSP pins are available for use, as noted in the table below. The voltage meter pin (pin 1/6) is pretty much dedicated for that use because of the voltage divider.

Multimeter – ADC, I/O, with resistor divider. (PIN2, PORTA0/AN0)

MCLR – input only pin. Schmitt trigger input for noisy signals. (PIN1, RE3)

Vcc - +3 volts

Vss – ground pin

Data – Input/Output with interrupt on change, optional weak pull-up (PIN27, RB6)

Clock – I/O with interrupt on change, optional weak pull-up (PIN28, RB7)

Firmware

Firmware was written using mikroBasic freeware version. Current firmware is v0.1. Future firmwares will probably be written in C.

Configuration options are set in the firmware. They should be as follows:

MCLR - DISABLED

BODEN/BOREN - DISABLED

WDT - DISABLED

Oscillator - Internal Osc, NO clock-out.

I was unable to program the 16F913 with my favorite PIC programming software (WinPIC800), but DL4YHS' WinPIC worked great [<http://www.qsl.net/dl4yhf/winpicpr.html>].

v0.1

Configuration/Menu System - Menu options scroll across the display and are selected/advanced using the two input buttons.

Time – displays the time in binary (default when a button is pressed).

Klik – a counter. I, on occasion, find myself performing counts. Traffic counts, bird counts, whatever. The *01/Vatch* subs as a binary counter.

Club Mode - The real value of any watch is determined by its 'club' mode. The *01/Vatch* uses a random number generator to flash patterns on the LED display. Its also possible to include word fragments using the internal matrix font library (more to come). Speed can be adjusted with button 1. The ultimate club upgrade package would include a temperature sensor that controls the rate of pattern change. As the wearer heats up, the patterns change faster.

Volt – voltage meter. Currently shows the raw ADC reading in 10 bits. Will be upgraded to actual volt value in v0.2.

Set – Set time.

Exit – Exit menu, put PIC in sleep mode.

Scrolling Menu System

Functions are accessed through the scrolling menu system. Menu items are loaded as bitmaps in an array and continuously scroll "upwards". Scroll is based on a multiple of the Timer0 mux driver. The scrolling menu "times out" using a multiple of Timer1 (seconds counter) after about 10 seconds.

Menu Options (Using the Watch)

(This applies to firmware version 0.1)

When a new battery is put in the watch it displays the 'SET' menu option by default. Touch button 2 to enter the set mode. The current time will be displayed (12:11). Use button 1 to increment hours, touch button 2 to advance to the next time unit (hours, 10's minutes, minutes). Touch button 2 after the minutes are set to save the time and return to the scrolling menu.

To save power, the display and PIC are usually off. Touch button 1 to wake the PIC and display the current time for 10 seconds. Touch button 2 while the time is displayed to access the scrolling menu system. The watch features are accessible through the scrolling menu. Touch button 1 to advance to the next menu item, touch button 2 to choose a menu item.

Button functions for each menu option are outlined in the table below. B1 and B2 are abbreviations of button 1 and button 2.

[SLEEP] PIC is sleeping, wakes occasionally to count seconds. Low power mode.

B1->Display time for 10 seconds.

[TIME] B1 wake the watch and display time for 10 seconds.

B2->Open scrolling menu.

[Menu] Press B2 during the time display to open the scrolling menu system.

B1-> Advance to the next option.

B2-> Select the current option.

[Time] Display time for 10 seconds, return to menu.

B2->Return to Menu

[Klik] Binary Counter (10 bit).

B1->Count (+1)

B2->Exit

[Club] Random pattern display (with adjustable speed).

B1->Change Speed.(5 levels)

B2->Exit

[Volt] Voltage Meter with precision reference (0~30 volts).

B1->N/A

B2->Exit

[Set] Set time.

B1->Increment Digit (set time)

B2->Next Digit (save current digit advance to next digit, save and exit after 3rd digit is set)

[X] Exit Menu

Firmware Roadmap

v0.2

An Exit Confirmation/Dialog.

Setup - Expand setup options to include:

- On-time duration/menu time-out (and an always-on mode).
- Brightness (duty cycle).
- Scroll Speed.

Menu Font Upgrade

- 'E' and 'B' look really bad, use 'e', 'b'.

Move to 1Mhz or 32.768khz oscillator (4MHz in v0.1).

v0.3

Stopwatch (time increment forward)

- Starts counting seconds, then increments minutes and hours after the 15:59 display limit.

Timer/Alarm (time increment backwards)

- A deincrementing timer, all LEDs flash when timer reaches 0.

EEPROM (logging values to flash memory)

- Save voltages, counts, options, stopwatch times, etc to flash EEPROM memory.

- Log number of days run since battery change. Also: number of hours with display on.

v0.4

External hardware features (using ICSP header):

- Event logging on interrupt.
- Bicycle Odometer/Speedometer.

Adjustable Unit Display (binary or decimal-font).

PCB

PCB and circuit are in the eagle format. I also included a bunch of libraries I used to make the board that may be needed.

The PCB is designed with mostly surface mount components. The board was made with inkjet transparencies on a photo positive board. This was my first surface mount board (both etch and assembly). I made a single sided board and used jumper wires for the bottom layer traces. The board was made with manufacture by Olimex in mind, so their 10mill rule check file was used when designing the board. Nothing is terribly small, but its certainly challenging. Everything was hand soldered using a 10 euro iron, stickie-tack, and a bright light. A magnifying glass was not needed.

The crystal was left as a surface mount component. The metal can is a distinctive looking element, and much more identifiable than a surface mount black-box. The prototype in the picture also uses a TO-92 voltage reference – the final PCB indicates a SOT-23 version that I didn't (yet) have on hand when I made the board.

Circuit and PCB are in the project archive (Cadsoft Eagle format – freeware version www.cadsoft.de). Component placement can be seen in the PCB file. I also made a PDF with the top layer mirrored and copied several times. This should be ready for toner transfer or foto process.

Parts list (through hole)

32.768kHz Watch Crystal (0206 metal can)

Pin header -x4
Programming header – 6 pins

Parts list (surface mount)

SO-300 PIC16F913/6
1206 0.1uF capacitor
1206 33pf capacitors - x2
1206 LED (yellow, red, orange, etc) -x12
1206 Resistor - 4x56 ohms
1206 Resistor - 3x1Kohm
1206 Resistor – 3x10Kohm
1206 Resistor - 3x100Kohm
SOT-23 NPN transistor (100ma or more)
SOT-23 PNP transistor (general purpose)
SOT-23 NPN Darlington transistor (general purpose, hfe of ~10000)
SOT-23 MCP1525 Voltage Reference (2.5 volts)
Battery CR2032 3v lithium

Potting the watch

To make the watch suitable for everyday use it needed a case. I visited AFF Materials [<http://www.aff-materials.com/>] to buy polyester resin. A nice guy there suggested that I use a clear epoxy instead. According to him, the polyester resin shrinks ~5% which might fracture connections on the PCB. The clear epoxy only shrinks ~2%. He also suggested that gases from the polyester might damage components while it cured.

Having never worked with a clear epoxy before, I did some test castings. I started by casting some samples in an ice cube tray. Sunflower seed oil, silicone lubricant, and silicone bicycle lubricant were tested as release agents. One sample was done with no release agent. The silicone lubricants beaded in the bottom of the mold and left pock marks on the epoxy. The control stuck to the bottom of the mold. The oil worked pretty well, but left a slight residue in the epoxy.

Next, I needed to know how to do a multi-layer casting with this material. A polyester resin is usually poured in layers. A first layer is allowed to set (about 15 minutes) to a gel. An object is placed on the first layer and a second layer of fresh resin is poured on top. The working time of my epoxy is about 60 minutes. I poured a first layer and checked it after 30 minutes – still soft. After about 1 hour and 15 minutes the first layer had stiffened enough to place an object on it. For this test I put the LED test board seen in step 2 face down on the first layer, and covered with a layer of fresh epoxy.

This worked great, the LEDs didn't pop off the board. I concluded here that absent a proper mold, the clearest surface I can make is the air/epoxy interface. The 'top' of the casting has a significant miscalculation. The miscalculation is limited to the very edge of the casing and is easily removed with a grinder.

For the first real test I needed a rectangular plastic mold. The best option I found was a 'smeer kaas' container. It wasn't perfect, so I made it smaller with a few layers of tape-wrapped foamcore. This wasn't a stellar mold, but choosing the top as the display surface gave me some leeway. The mold was lightly wiped with oil on a paper towel.

I ditched the multi-layer pour procedure from above. I soldered leads from the coin cell battery holder

to the PCB. The cell holder was hot-glued (ok, stickie-tacked) to the bottom of the PCB. The battery holder was filled with stickie-tack, and the programming header protected with yet more stickie tack (plasticine would also work great). This was then placed, face up, in the mold. The stickie tack protecting the battery and header was pressed firmly into the bottom of the mold, anchoring the watch in place. Clear epoxy was poured into the mold until it covered the watch. The pin-headers were still quite long, but can be cut after the epoxy dries.

The watch released from the mold after about 36 hours. The protective putty was removed with a screw-driver. The edges were smoothed with a drill-press grinder bit. The watch was cast a little large to be worn as a wrist watch. I may try to cut it down if I can find a band saw. For the time being, it will be a pocket watch. The tape-over-foamcore gave a cool texture and ultra-clear surface. Next time I will try to make the entire mold using this material, something more in the neighborhood of wrist watch size.

Improvements

In addition to the software updates outlined in the roadmap, there are several additional areas for improvement.

Hardware

A 4x5 matrix of 0805 LEDs would take up the same space as the existing 1206 array. I purchased several types of 0805 LEDs to try in future designs.

The previously mentioned temperature sensor could be added to make an advanced 'club-mode' upgrade package.

The PCB was designed for manufacturer by Olimex as a double sided board (~\$33). They work straight from Eagle files and panelize (make multiple smaller boards from one big board) for free. I haven't done this, but I would buy one if someone else had them made.

Software

The 16F913/6 have a ton of space. A speedometer/odometer is planned. Games could be added.