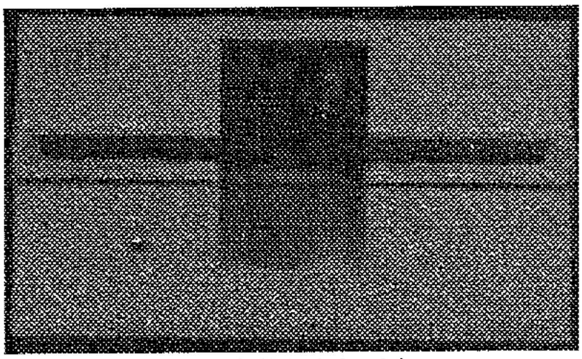
## Using the motor from an original IBM 5 1/4 inch floppy drive for the propeller clock

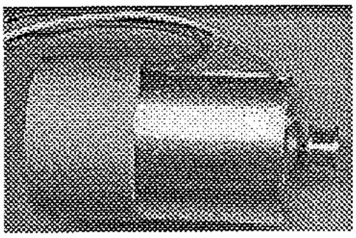
Lots of you will will not be able to find the Sharp vcr reel motor that I used for the propeller clock. I've found another motor you can use, and figured out how to modify it to use in the clock.

You'll need to find a Tandon full-height 5 1/4 inch floppy drive, such as used in the original IBM PC and many clones. Other manufacturers made similar looking drives, but Tandon made the best ones, with a motor made by Buehler. That's the one you want.

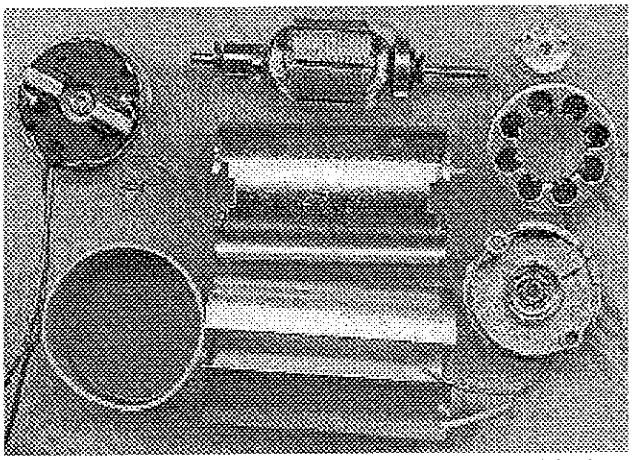
This motor runs a lot faster than the Sharp motor. The Sharp motor runs about 1800 rpm with 6.5 volts in, and this motor runs about 3000 rpm. You'll probably want to run it at 4.5 to 5 volts to slow it down. The side-effect of this is decreased brightness of your clock(remember, the clock uses the motor voltage minus about 1.25 volts). Use high-brightness LEDs and 47 ohm resistors instead of 120 ohms.



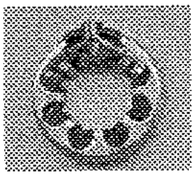
A typical Tandon full-height drive.



The Buehler motor.



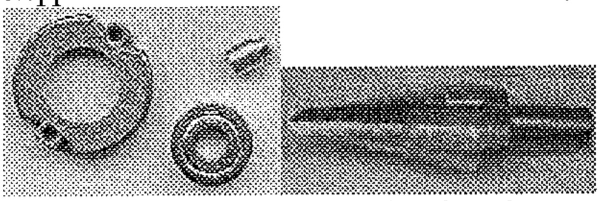
Motor after disassembly. The brush assembly has two small brushes and two small springs that love to fly. You might want to do some of the disassembly inside a clear plastic bag.



Discard the frequency generator assembly.

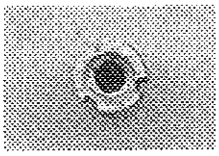
---

Disassemble the rest of the disk drive, in particular the main platter/flywheel assembly. It contains two bearings. Save the flanged bearing. It measures 0.625"OD and 0.25"ID.
Drill a 0.625" hole in the motor's end cap(the nose end, not the end with the brush assembly). A stepped drill bit does a nice job.
Get a 0.25" round threaded spacer. The disk drive does not contain one, although the main platter shaft might tempt you. It is probably harder than your hacksaw blade, so forget it. The spacer shown here is stepped down at one end. This is not needed, it was the only type I had with the right dimensions.



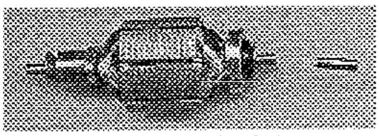
The end cap after drilling, the bearing, the spacer, and a step drill

---



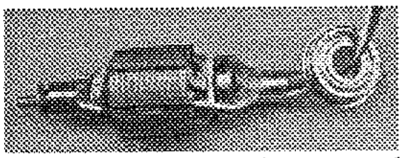
Use a hacksaw to cut three notches in the spacer. The notches should run from end to end, and be big enough to fit a small insulated wire. This is an enlarged picture showing one end of the spacer.

---



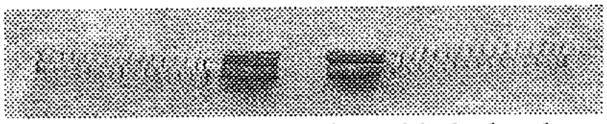
Grind off the end of the motor shaft, leaving about 0.2". Don't ruin your hacksaw blade trying to saw the shaft.

---



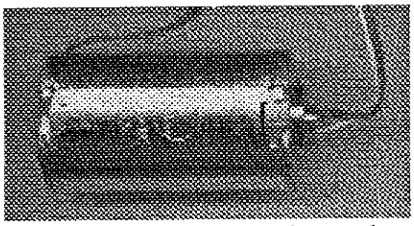
Glue the spacer on the motor shaft and let dry. I used "Automotive Goop", but you might also have success with "J.B. Weld" or epoxy. Clean all parts with alcohol before gluing.
You'll need three small insulated wires with a Berg connector on one end. The disk drive has lots of wires that are too big to squeeze into the notches in your spacer. The coaxial head wires will work if you strip off the outer insulation and shielding.
Thread the wires through the ball bearing. The bearing flange will need to be on the inside of the motor.
Solder the three wires to terminals on the motor's commutator. The commutator has 7 terminals, you have 3 wires, so symmetry is not possible. Best performance will be with two of the wires on adjacent terminals, and the third wire on the opposite side. Do not disturb the existing motor wires.
Glue the wires to the motor, embedding them in the spaces between armature sections and tying them tightly with thread or dental floss.

---



Assemble the motor, starting with the brush assembly. The motor brushes need to be installed correctly, the curved ends towards the center, and the slanted ends to the springs. The grooves on the brushes should be visible as you assemble the brush assembly. To catch flying parts, assemble the brush assembly inside a clear plastic bag.

---



The completed motor has a threaded shaft to mount the propeller clock to, and a three-wire connector that rotates with the motor to attach to the clock circuit.

---

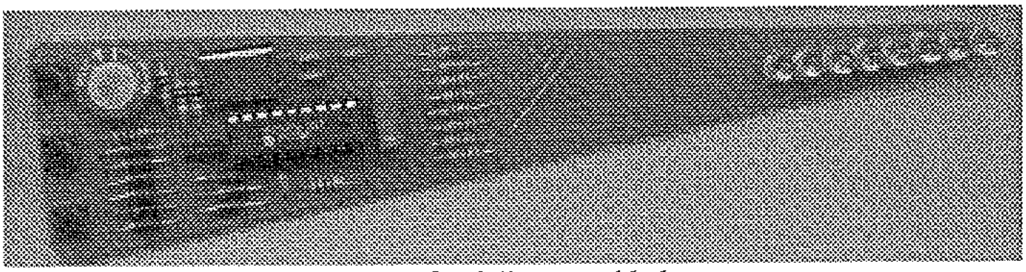
Return to the Propeller Clock main page.

---

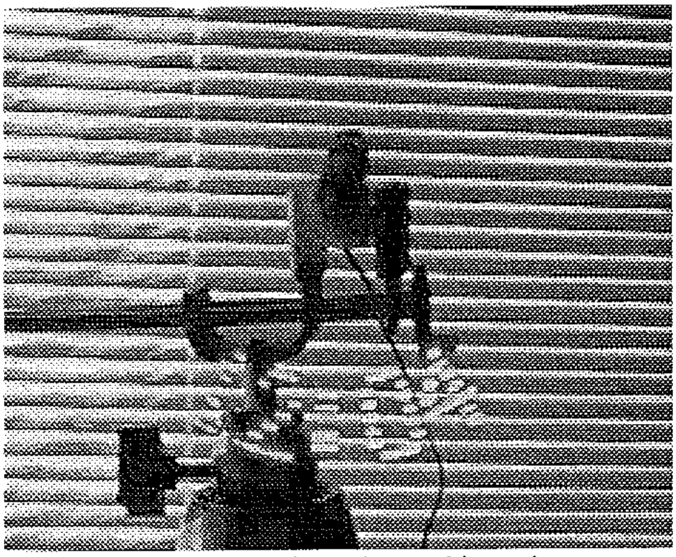
**A printed circuit board(PCB) layout for the propeller clock.**

If it's easy for you to make printed circuit boards, here's a layout you may use for your clock. It's all one piece, sort of a "wall model". This upcoming winter I will most likely write a new version of the code that will simulate "analog clock hands", and this model will work nicely with that.

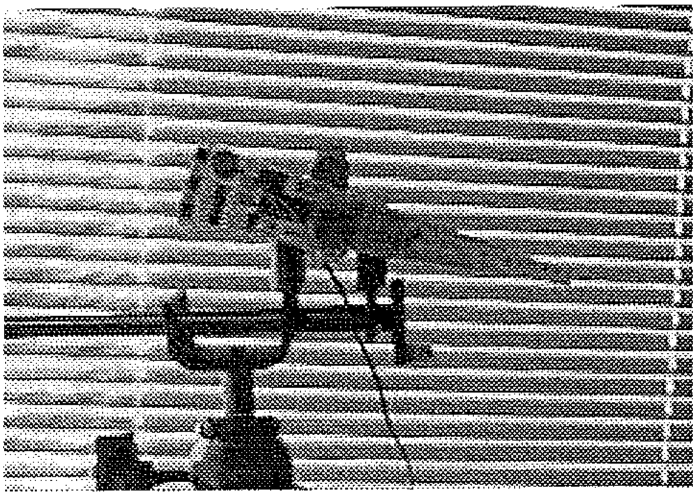
This layout is for personal use only, please do not use this design to manufacture them for sale.



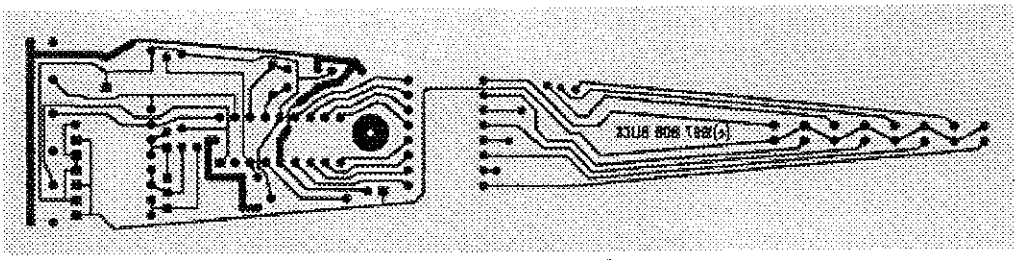
The PCB fully assembled.



The clock running, clamped in a vise.



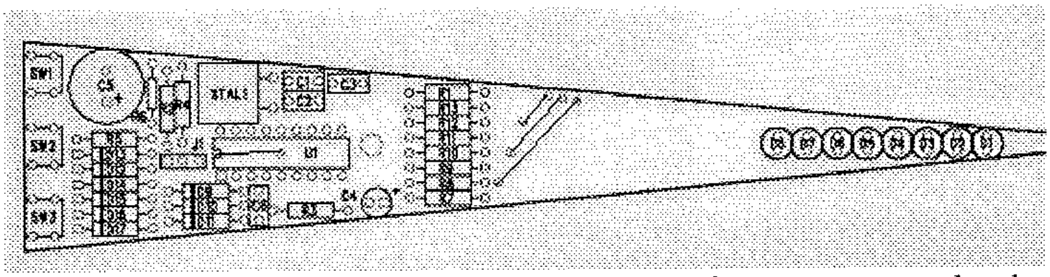
Same thing, not running.



Foil view of the PCB.

To make a laser printed transfer, use one of these two files that I created "print to disk". You'll need to drop to DOS and "copy filename lpt1" in order to print. Remember to substitute the real filename for "filename" and the correct printer port if it isn't lpt1. If you have a Macintosh, you'll have to find one of those utilities that allow you to copy a file straight to a printer with no translation. If you have something besides a Mac or PC, you're probably smart enough to figure it out without my help.
clkfoil.prn HP Laserjet IIP format.
clkfoil.ps Postscript format.

---



Here's a stuffing diagram for the propeller clock PC board. Part references correspond to the parts list below. **Important note:** There are four wire jumpers on the board. One is under U1 IC socket.

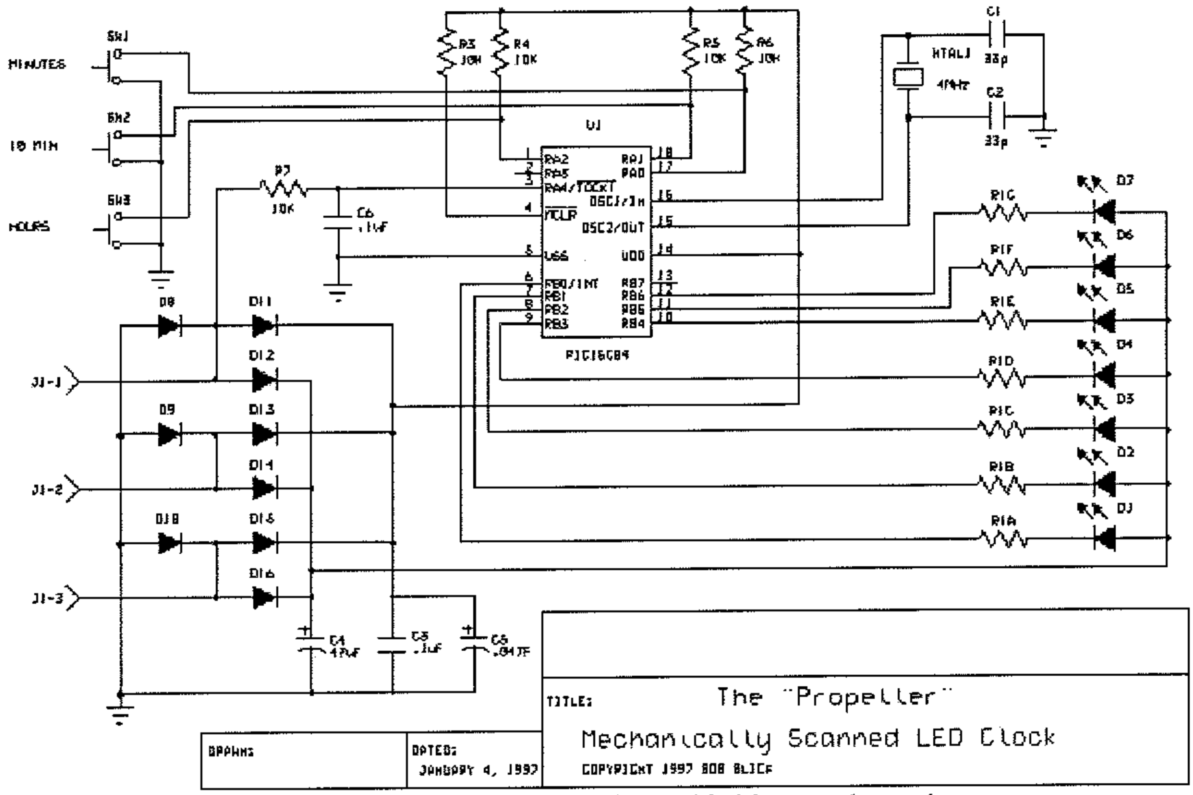
```
The Propeller Clock Parts List(matches PC board)

Capacitors:
C1, C2 - 15-33pf ceramic(not variable, just not critical)
C3, C6 - 0.1uf ceramic
C4 - 47uf electrolytic
C5 - 0.47 farad (47,000uf) supercap(memory cap)

Diodes:
D1-D7 - light emitting diodes. Use bright ones that cost a lot.
D8 - Optional LED for the new version I'm doing this winter.
D9-16 - 1N4001 general purpose 1 amp rectifiers

Resistors:
R1,R7-R13 - 120 to 220 ohm. Use 100 ohms if you dare. I do.
R2-R6 - 10k ohm

Misc:
J1 - three terminal Berg connector
SW1-SW3 - normally open pushbutton switches
U1 - PIC16C84 or PIC16F84 programmed with mclock code.
XTAL1 - 4MHz crystal
IC socket for U1, 18 pins.
```

---

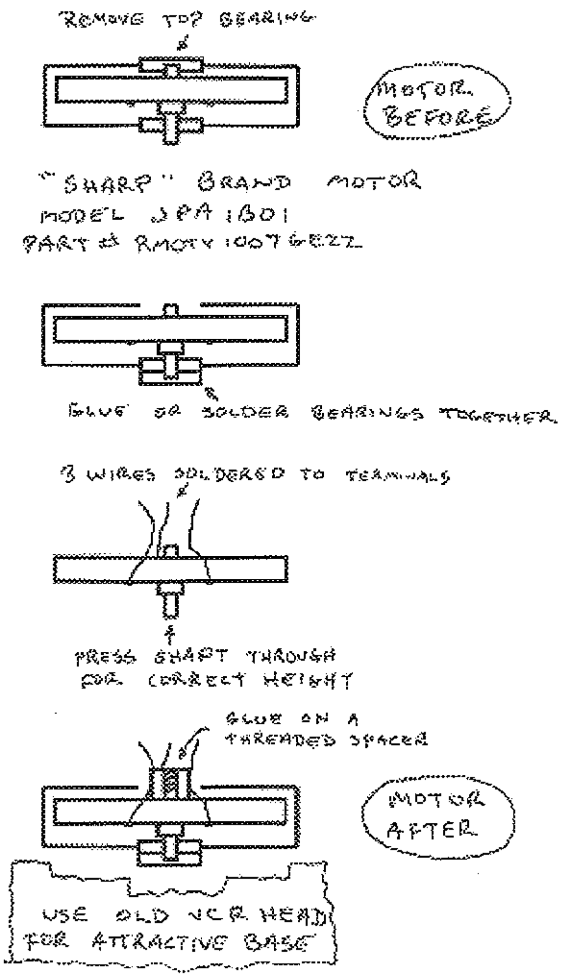
Return to the Propeller Clock main page.

This page created September 7, 1997 and likely contains errors. Send me email to bob@ted.net with corrections if you have any.

---




This schematic is hard to read, get this bigger schematic.




## The Propeller Clock Parts List

**Capacitors:**
C1, C2 - 33pf ceramic
C3, C6 - 0.1uf ceramic
C4 - 47uf electrolytic
C5 - 47,000uf supercap(memory cap)

**Diodes:**
D1-D7 - light emitting diodes
D8-16 - 1N4001 general purpose 1 amp rectifiers

**Resistors:**
R1 - 120 ohm DIP array or seven 120 ohm resistors
R2-R6 - 10k ohm

**Misc:**
J1 - three terminal Berg connector
SW1-SW3 - normally open pushbutton switches
U1 - PIC16C84 programmed with mclock code
XTAL1 - 4MHz crystal
MOTOR - Sharp RMOTV1007GEZZ

---

**The Propeller, a mechanically scanned LED clock by Bob Blick.**

The clock is on a spinning piece of perfboard, but it must get power. I thought of many ways to do this, including using two motors(motor one has its shaft fixed to a base, and motor two spins the body of motor one, generating electricity), making a rotary transformer, or using slip rings.

I decided to do it another way, taking power from the spinning armature of a plain DC motor. In order to run the wires out of the motor, I removed the bearing from one end of the motor, leaving a big hole.

There are three terminals inside most small DC motors, and it acts a lot like three-phase alternating current, so it must be rectified back to DC. A nice side effect of this is that the position of the motor can be detected by taking one of the phases straight into the microprocessor.

**Step One: Mangle a Motor.**

Find a VCR, perhaps a Sharp or a Samsung, with a flat reel motor. The motor I have is marked JPA1B01, but Sharp knows it by the number RMOTV1007GEZZ. Take it apart without mangling the brushes(there are little holes to slip a paperclip into to move the brushes out of the way), and notice that it has one ball bearing and one sleeve bearing. Knock the sleeve bearing out of the case and glue or solder it to the other end of the motor, as an extension of the ball bearing. The shaft of the motor will have to be repositioned slightly to get the right height, press it in a vise with a hollow spacer on one end. Take a Berg connector with three wires and solder them to the three terminals on the motor's armature. Glue a short threaded spacer to the shaft at the end that will stick out the hole, and reassemble the motor(be careful with the brushes). You can glue the motor to a VCR head as a weighted base.

**Step Two: Build the circuit.**

I used perfboard(Vectorboard) and handwired the circuit together. Use an 18-pin socket for the 16C84 because it needs to be programmed before putting it in the circuit. For the 7 current-limit resistors I used a DIP resistor array, because it made it easy to experiment with LED brightness. I settled on 120 ohms. You can use seven regular resistors, because 120 ohms works fine, though it puts the peak current right at the limit for the 16C84. Think about balance while you build this circuit, and reference my pictures, so you don't have to add a lot of balancing weight later. Substitute for any part values you like. Note that I used a 47000uf supercap, it is to keep the clock running after turning it off, so you can set the time. The LEDs get power separate from this. Don't substitute a ceramic resonator for the 4MHz crystal, this is a clock and should be accurate.

### Step Three: Program the 16C84.

You'll need a programmer that will program a PIC16C84. If you found this file/web page, you can find plans to build a 16C84 programmer. Program it using the hex file accompanying this document. I have included the source code(.asm) just for your amusement. When programming the chip, set the chip options to: watchdog timer(WDT) OFF and oscillator to normal XT crystal.

### Step Four: Throw It Together and Keep Time.

Screw the circuit board to the motor, and plug the three wire connector in. Apply power to the motor. The preferred voltage is 6.2 volts, but it will run from 5 volts to about 7.5 volts. Note that 5 volts gets to the circuit when 6.2 volts is applied to the motor, because of diode losses. The clock may be working at this point, displaying 12:00. If it isn't. there was probably some voltage on the supercap when you plugged in the chip. Turn off the power and momentarily short pins 5 and 4 together(ground and /mclr) to reset the chip. Now when you apply power the clock should work, and you can set it by turning off the power and pushing the buttons(hours, 10 minutes, minutes) the right number of times. If the numbers appear backwards, reverse the polarity to the motor to make it spin the other way. You might experiment with balancing the clock, and the use of foam under the base to reduce vibration.

### Step Five: Modifications.

If you look closely at the source code, you'll see that the "dot rate" is adjusted to the speed of the motor to make the display a consistent width regardless of the motor's speed. The motor I used has brushes set 90 degrees apart, and gives two indexes each revolution. The clock displays on two sides, 180 degrees apart. If you use a motor with the brushes 180 apart, the clock will only display on one side, and the numbers will be too wide. You'll want to modify the program, in the section marked D_lookup_3. The value in the W register when Delay gets called effects the width of the digits. You might try sending half of the period_calc value to Delay, perhaps by rotating period_calc right into W(remember to clear the carry flag first). Like this:

```
bcf STATUS,C
rrf period_calc,w
call Delay
```

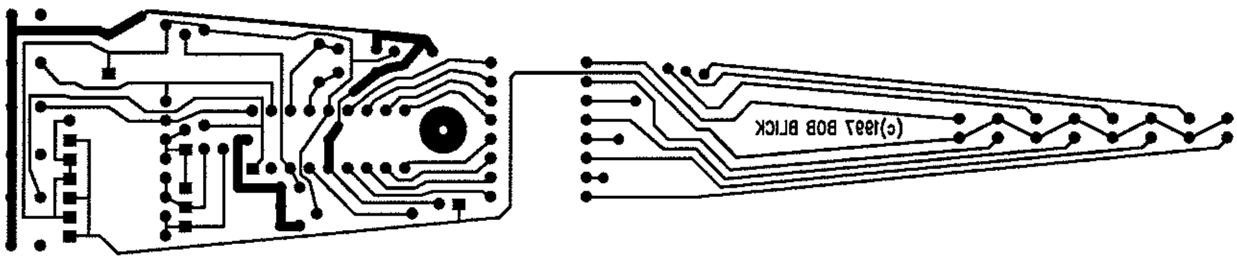
January 25, 1997 Bob Blick
(bob@ted.net, bob@bobblick.com)

You can email me bob@ted.net

---

Go back to mclock page 1.
Go to my home page.

---


(c)1997 BOB BLICK