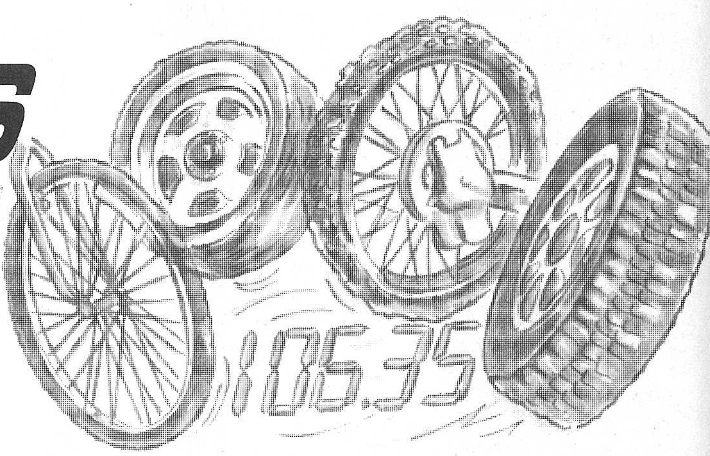
PIC-AGORAS WHEELIE METER



JOHN BECKER

Part 2

Keep a check on your speed and distance. From child's tricycle to heavy-haulage vehicle.

LAST month we introduced the megnetic field sensor and dealt with the circuits and construction of the frequency-to-logic interface, display and processing stages. The system software demands were also discussed.

We conclude this month with the first display run and possible downloading failures. Also, details of setting the wheel size and possible alternative sensing are covered.

FIRST DISPLAY RUN

Switching off S3 (Reset), the program inside the PIC should now start running, though preset VR2 will probably need adjusting before anything is seen on the l.c.d. screen. What you should see is Line 1 showing the time counting upwards in seconds, from a zero starting position which you may have missed while adjusting VR2. Line 2 should show "KD000.00".

Rotate switch S4 through all eight Mode positions, observing that the display shows what appears to be sensible information (see Part 1 for typical Mode displays), and that the buzzer sounds when in Mode 7.

It is just possible, particularly if the PIC has been used before, that non-zero results may appear in some lines for which the data should at present be zero. This situation can be resolved later by initiating the Full Reset option.

DOWNLOADING FAILURE

This section will be of particular interest to anyone who has had problems programming PICs.

Apart from errors in assembly (or the unlikely event of a faulty component), there are two reasons why the program may not behave as expected, probably resulting in no screen display or one that is unexpected.

Sometimes, if the leads to the computer are a bit long, the data being transferred may become corrupted. Re-send (download) the data and try again. Shorten the leads if necessary.

The other possibility is that the computer has been configured to output information to the printer port via a different register than the downloading software has been set for. The TASM Send software is set to output via the "normal" register &H378.

To check the printer port operation, run the following test program from QBasic or GWBasic:

```
10 FOR A = 0 TO 255
20 OUT &H378, A
30 NEXT A
40 GOTO 10
```

This toggles the printer register &H378, and thus the printer port lines if this register is the one the computer has been configured to use. An oscilloscope can be used to monitor connector PL2 pins 2 and 3 (port lines DA0 and DA1) to check that they are toggling between high and low. Line DA1 should toggle at half the rate of DA0.

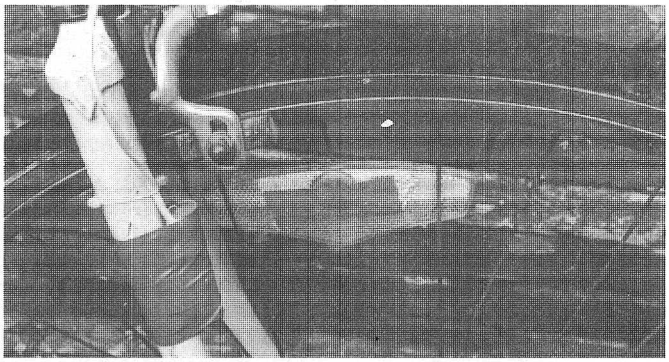
A multimeter (preferably analogue) can be used if a scope is not available. Set it to a suitable range for +5V d.c. and observe the meter display. It will probably be necessary to insert a delay loop in the above program so that the meter readings can be seen more slowly. The following is a good starting point (inserting it between lines 20 and 30) :

```
25 FOR B = 1 TO 10000: NEXT B
```

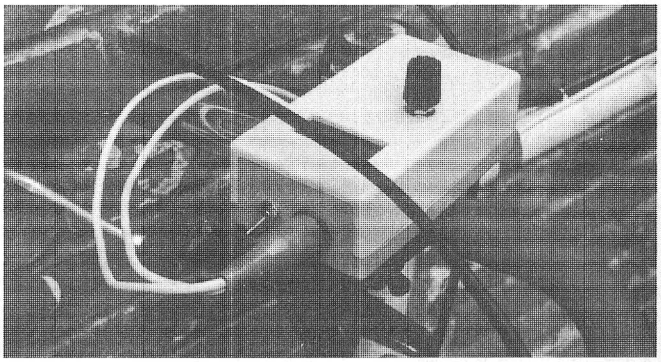
The value of 10000 can be increased if the delay is still not long enough.

If it is found that the port register is not &H378, the computer must have its printer configuration changed. You should consult your computer manual to find out how this is done. The TASM Send program cannot be modified except by programmers experienced in C and having the necessary compilers.

A related problem may be that attempts are made to assemble/download the data which is in the wrong format for the downloading program. TASM files need TASM downloading software, for example.



The magnetic field sensor taped to the bike fork and the magnet glued and taped to a plastic reflective plate attached to the wheel spokes.



The display/control unit clamped to the bike handlebars. The two leads from the DIN plug go to the sensor and battery (located in rear saddle-bag).

SETTING WHEEL SIZE

Readers who will be doing their own programming of the PIC16C84 may prefer to set the wheel diameter size directly into the software, using the modified program referred to earlier. Those of you who are using pre-programmed chips, or unmodified software, should still read the calculation aspects of this section as you still need to know how to establish the parameters to suit your wheel.

For the *internally preset* method, wheel diameter details are held in two sub-routine calls whose details need to be entered before the PIC is programmed (if preferred, they can be changed later and the PIC re-programmed).

About 270 lines into the ASCII source code text file are the two statements:

```
SIZE1: retlw 45 ;lsb wheel size 27.5 inches
SIZE2: retlw 56 ;msb wheel size 27.5 inches
```

When these routines are called, the software returns to the calling point with the stated value (45 or 56) held in its accumulator register.

The values shown are those calculated for the author's wheel size of 27·5 inches diameter. Other values can be calculated through the following QBasic/GWBasic program, which uses the diameter of 27·5 inches as the example:

```
10 DIA = 27.5: PI = 3.1416: MM = 25.4
20 REM MM = 25.4 only for wheel diameter in inches
30 REM if wheel diameter is in millimetres, set MM = 1
40 CIRCUM = DIA * PI * MM
50 X = INT(65536 * CIRCUM / 10000)
60 MSB = INT(X / 256)
70 LSB = X – (MSB * 256)
80 CLS: LOCATE 10, 1
90 PRINT "MSB ="; MSB; "LSB ="; LSB
```

(Those who wish to use a hand calculator should note that **INT** means *integer*, i.e. ignore the fraction.)

The screen – displayed answer in this case is:

MSB = 56 LSB = 45

From this answer, 56 is the SIZE2 value, and 45 is the SIZE1 value.

If the wheel size is in millimetres, change the value of MM in Line 10 to a value of 1.

The statements following the semicolon (;) in the SIZE1 and SIZE2 sub-routine lines do not need to be changed, except for your own information; the software ignores anything following a semicolon.

Theoretically, the smallest diameter wheel that can be used in calculation is 0·002 inches (0·0508mm) (m.s.b. = 0, l.s.b. = 1), though you might have problems mounting the magnet! The maximum wheel diameter that can be monitored is about 125·317 inches (3183·052mm) (m.s.b. = 255, l.s.b. = 255).

When considering the diameter of a wheel having pneumatic tyres, calculated answers may never really be borne out in practice since the tyres of a bicycle, for example, will flatten slightly when the rider mounts the bike, so changing the actual distance covered per wheel revolution. Different rider weights will have different flattening effects! This situation applies to any distance/rotation calculator used with flexible tyres.

The maximum speed that can be detected for any wheel depends on its size. As previous stated, the maximum wheel rotation rate that can be used is 25Hz. Therefore, for a 27·5 inch diameter wheel, the maximum speed is:

diameter in inches × Π × 25Hz × seconds per hour / inches per mile

Thus: 27·5 × 3·1416 × 25 × 3600 / 63360 = 120·49 m.p.h. (192·78 k.p.h.).

The 25Hz limit is dictated by three factors, the rate at which data can be processed, the number of memory bits allocated to calculations, and the *RC* time constant of resistor R4 and capacitor C2 on the output of IC1.

If the maximum pulse rate is exceeded, the velocity line in Mode 1 and Mode 4 will have the word MAX substituted for the speed value. In Mode 6, the TA value will show 255.

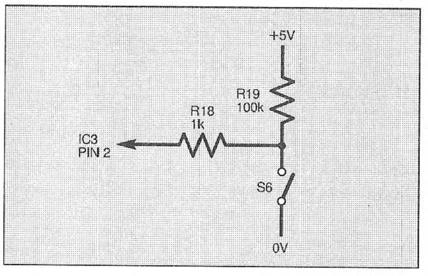


Fig. 5. Circuit for triggering wheel diameter parameters into PIC16C84 EEPROM Data Memory.

EXTERNAL SETTING

With the unmodified software and pre-programmed chips, wheel size data is entered by using a signal generator. Three components also need to be temporarily added, as shown in Fig. 5. Disconnect the wires of the buzzer, WD1. Switch off the additional switch, S6. Calculate the values of the m.s.b. and l.s.b. for your wheel diameter.

Connect the signal generator to test point TP1. Switch on PIC-Agoras and the signal generator, set to a 5V square wave output at between 0·1Hz and 25Hz. Switch to Mode 6, and observe line 2 (display 'TA xxx').

Adjust the signal generator until line 2 displays the m.s.b. value. Since the display shows an average of the pulse count received over 10 seconds, allow at least that time to elapse between slight adjustments to the signal frequency.

When the correctly displayed m.s.b. value has stabilised, switch to Mode 0 (showing time and kilometres distance), then switch on the new switch S6, leave it on for a couple of seconds, and then switch it off. The m.s.b. value is now programmed into the PIC's EEPROM data memory.

Switch back to Mode 6 and adjust the signal frequency so that line 2 now displays the wheel's l.s.b. value. Allow it to stabilise, then switch to Mode 1 (showing kilometres speed and average). Again switch on the new switch S6, leave it on for a couple of seconds, and then switch it off. The l.s.b. value is now programmed into the PIC.

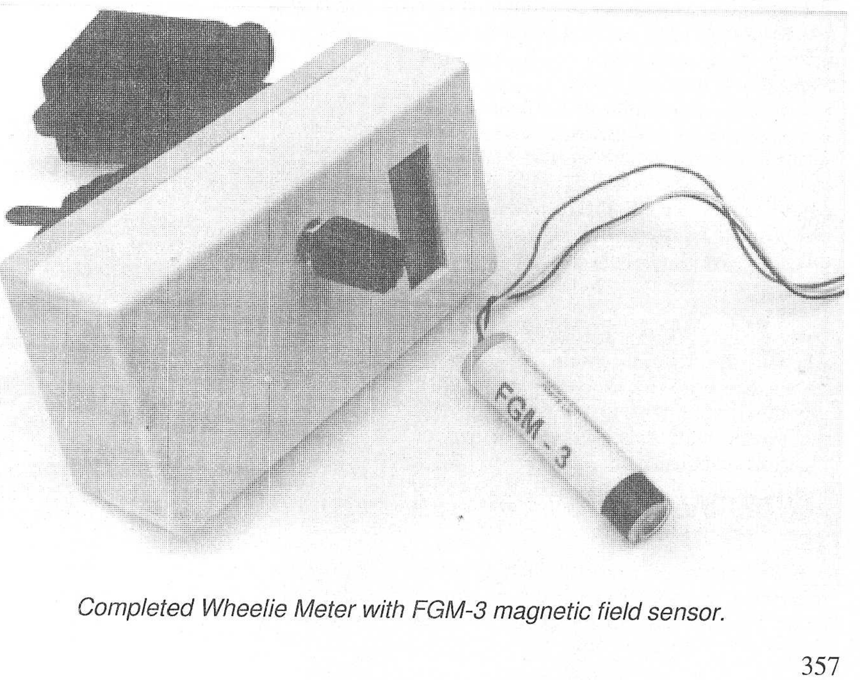
Programming space available (three bytes left!) has not allowed room for a routine to provide a visual check of the EEPROM's new contents, but you could verify them indirectly by setting the signal generator to different frequencies for which you have calculated the equivalent factors for speed and distance, as if the pulses were being generated by the wheel itself.

When satisfied, disconnect the three temporary components and the signal generator, and reconnect the buzzer. That's all there is to the external programming. The same technique can be used as many times as you want to change the wheel diameter factors.

SIGNAL GENERATOR

Most electronics enthusiasts should have a signal generator, but if you don't, you can construct a suitable one on a breadboard using the circuit in Fig. 6. No assembly details are offered, but stripboard is the suggested constructional base.

Potentiometers VR3 and VR4 provide Coarse and Fine adjustment respectively.



Completed Wheelie Meter with FGM-3 magnetic field sensor.

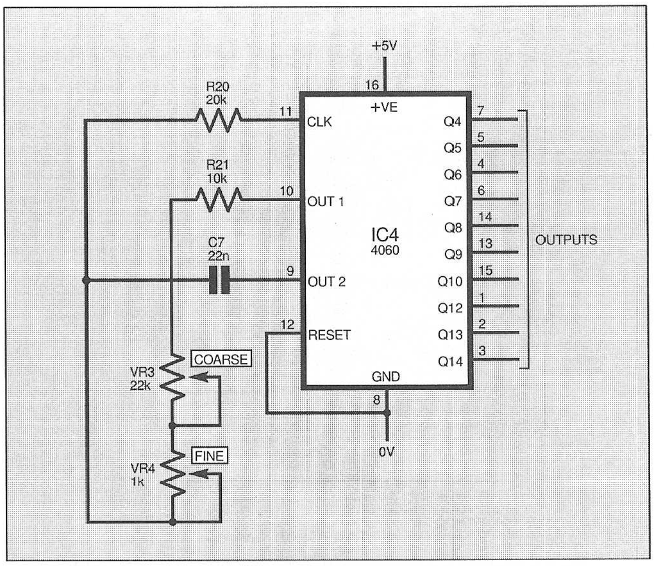
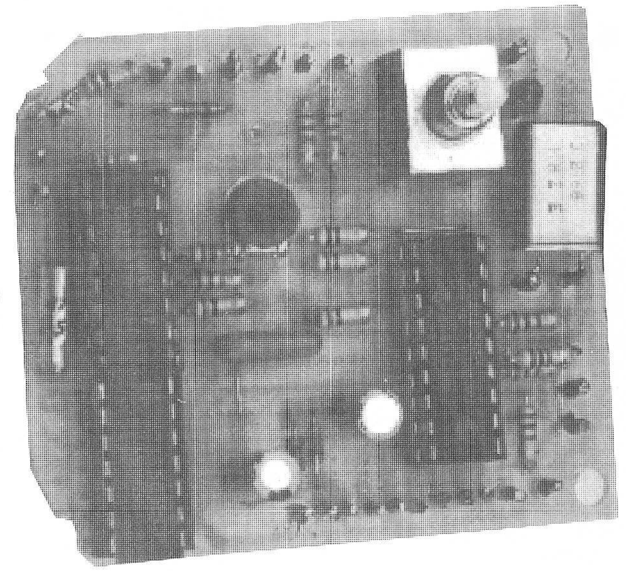


Fig. 6. Circuit for a simple Squarewave Signal Generator.



Completed prototype control board for the distance/speed monitor.

The signal can be taken from any of the output pins. The value of capacitor C7 may be changed if you want to alter the frequency range for another application.

A breadboarded version of this generator produced a basic range of 750Hz to 2000Hz (at pin 9). The output range at Q6 was thus 11·72Hz to 31·25Hz, and at Q14 it was 0·05Hz to 0·12Hz.

MOUNTING - UP

On the author's bike, the unit's case is bolted to the top of the bracket normally used for the bike's battery-powered front lamp (from Halford's). A suitable bolt already exists in the bracket. The battery is held in the pocket of a rear saddle-bag, the wires being run along the frame and secured with cable ties and insulating tape at convenient intervals.

The magnetic sensor X1 is secured to a front fork with insulating tape. Its wires are lengthened, using insulating tape to cover the connections (a 3-way connector block could be used), and then terminated by plug PL1.

The magnet is fixed to one of the clear plastic reflective plates which most bike's have on their wheels. Again, insulating tape can be used, or a strong glue.

It must be remembered, though, that fast wheel rotation rates will put a strain on the magnet fixing and unless really secure, it could fly off, causing injury.

Recalling the results of the bench tests with the magnet and sensor, position them relative to each other for the maximum change when the wheel is rotated. Then adjust preset VR1 until the best output response at IC2b pin 10 occurs. During this time, Mode switch S4 can be set to Mode 6 and the l.c.d.'s changing rotation count observed.

Note that PIC-Agoras does not know whether the bike is going forwards or backwards, or if you are stationary and just idly moving the wheel back and forth with the sensor and magnet at the fringe of high/low response.

ALTERNATIVE SENSING

Other types of sensor which produce individual pulses per wheel revolution could be used instead of the FGM-3. In this situation, IC1, the p.l.l., is not required and should be omitted, as should resistors R1 to R4, capacitors C1 and C2, and preset VR1.

The pulses from the alternative sensor can be brought in at test point TP1, using IC2a and IC2b as buffers. The pulses must have normal 5V logic levels.

For test purpose, signal generator pulses can also be brought into TP1, without removing IC1 or other components, should you want to trigger the microcontroller without using the sensor.

As a suggestion for experimentation, a wind speed sensor or even a water flow sensor could have its output fed into TP1, making appropriate changes to the programmed wheel size value to suit.

Perhaps even a speed indicator for boat or sail-board use could be evolved from this suggestion, as long as adequate waterproofing is ensured!

SOFTWARE SOURCES

Copies of the author's PIC program for this unit are available on 3·5-inch disk from the Editorial Office for the sum of £2·50 UK, £3·10 overseas surface mail or £4·10 airmail. This is to cover admin costs and postage. The disk itself is *free*.

The program can also be downloaded *free* from our FTP site: **ftp://ftp.epemag.wimborne.co.uk**, in the sub-directory: **pub/PICS/PICagoras**.

The source code is written for the TASM assembler, which is included on the same disk and also available from our FTP site. Experienced programmers can translate the code to suit other assemblers, such as MPASM, for example.

Magenta Electronics have offered to supply PIC16C84s pre-programmed with the software, see *Shop Talk*. The author is grateful to them for supplying him with an intelligent l.c.d. module for use in the development of this project.

WHAT'S IN A NAME?

PIC – Agoras? Why?

It was one of those days at the office when euphoria rolled through the atmosphere and banter exchange reached nearly warp speed. Queries Tech-Ed to Ed: "What can we call it? It's more than a bike computer, less than the infinite answer to wheeled life as we know it."

It's nearly *The Ultimate PIC-uppable Revolutionary Versatile Wheeled Distance Computer for Anyone Who's Going Places*. Bit long, perhaps", cogitates Ed. "Any Greek gods of the Wheel?"

"Helios and his chariot, that's got wheels; *HelioPIC?*"

"Would need to be solar powered."

"There's the mythical Cyclops; *CycloPIC*?"

"Just says *bikes for the optically impoverished*."

"Greek mathematicians?"

"Euclid; Archimedes; Aristotle; Aristweed; Pythagoras ..."

"Modern Greeks seem to call him *Pit – a – gor – rass*. His birth village still exists – on Samos, named after him – Πυθαγορειον. They've got a statue."

"PIC – Agoras! That's it – angles of the dangles; arc-eology; diametrics; distance round a wheel!"

And so the World revolves . . . □

Homage to Pythagoras (Πυθαγορας) on the Greek island of Samos.