```
'                    ****************************************
'                    *        Bicycle Odometer Program              *
'                    ****************************************
'************************************************************************
'*The program sends a 1 ms pulse to a counter every 100 metres. It is assumed that  *
'*the wheel diameter is 70 cm. The circumference is then 2.2 m. The bicycle travels  *
'*100 m in 100/2.2 = 45.5 rotations of the wheel. Each rotation is sensed by a reed *
'*switch which is activated by a magnet mounted on the spokes. Pulses from the reed *
'*switch are counted in software by the PicAxe program. Every 45 or 46 rotations, a  *
'*pulse is sent to a counter. This fixes the 45.5 turns problem. This calibration can be *
'*changed to suit other wheel diameters. Although the output of the PicAxe chip can *
'*be counted with a digital counter, another option is to use a cheap, small calculator  *
'* to perform the counting task. The calculator is initialised using a sequence of  *
'* commands from the outputs of the PicAxe chip. *
'************************************************************************
'
symbol counter1 = b0                    'set initial conditions for counter
symbol flag1 = b1                       'and flag1
b0 = 0
b1 = 0
main:
If pin3 =1 and flag1 = 0 then Inc1      'check for pulse from reed switch
If pin3 =1 and flag1 = 1 then Inc2      'if high go to Inc1 or Inc2 routine

goto main                               'if low cycle again

Inc1:                                   'counter for 0.70 m wheel diameter

counter1 = counter1 + 1                 'increment counter with each rotation
pulsout 2, 100                          'send 1 ms pulse to pin 2 of
                                        'microcontroller chip to reset flipflop
If counter1 = 45 then Puls              'test for 45 counts (~ 100 m distance)

goto main                               'return to main program

Inc2:                                   'counter for 0.70 m wheel diameter

counter1 = counter1 + 1                 'increment counter with each rotation
pulsout 2, 100                          'send 1 ms pulse to pin 2 of
                                        'microcontroller chip to reset flipflop
If counter1 = 46 then Puls              'test for 46 counts (~ 100 m distance)
goto main                               'return to main program

Puls:
pulsout 4, 100                          'send 1 ms pulse to pin 4 of the
                                        'microcontroller chip

b0 = 0                                  'reset counter1 to zero

If flag1 = 0 then flaga                 'conditionally reset flag
If flag1 = 1 then flagb

flaga:
flag1 =1
goto main
flagb:
flag1 = 0
goto main
```

'Notes: After counter1 is reset, a pulse is sent to pin2 which then resets the flipflop. The 'flipflop is also connected to the reed switch. This avoids retrigger from one wheel rotation.