Gauge ‘1’ Jigger

Two years ago I saw Glen Anthony’s (ChCh Garden Railway Society) jigger on his website. I was enthused. I must build one of those.
A couple of months before our ‘Model Mee” exhibition I read an article about a model that chased and slowed when approx 500mm from the model in front. It used Infrared reflections to keep its distance. It was controlled by a Picaxe microcontroller. I was sold on the idea as I am a Picaxe fan.

I approached Bruce Geange and we designed the chassis from 6mm square aluminium. I then built the top section from 2.3mm custom wood. The character driving is a very modified Flanders from the Flintstones series.
Parts were scrounged from the junk box and the drive motor from a discarded CDROM drive.
The control system was hurriedly put together but I could not get it to work reliably. The software and PC board was altered and a simple remote control was used. I used the remote from the Rudolf Picaxe (AXE 107) project.
The jigger now uses track power as long as it is greater than 6V. There is a 5V regulator to run the Picaxe and the motor. It has a Infrared receiver to give, Stop, Slow and Fast speeds.

The model performed extremely well on its maiden run at Palmerston North Model Engineering Clubs Model Mee 2006 exhibition and also at Grant and Donna Alexander’s Squirrel Valley Railroad, Cambridge. Below is the code for the Jigger and Remote Control.

```
'*****************************************************************
'                  Jigger with IR Control - PICAXE 08M
'
'         201 bytes used of 256  -  7 October 2006  -  Version 4
'*****************************************************************
'                           100nF     2000uF 4002 100nF      470uF
'          .--.--------------.----------.--|<-.--.---.--5VREG-.--.
'          |  |  PICAXE-08M  |          |     |  |   |        |  |
'     180 /  |              ===        ===    - === / \       |  |
'    ____ \  |  .---.---.   |    WhLed |     ^  | MOTOR      | |+|
'   |   1|-'  '->| 1   8 |<--^----------{     |  |  \ /      ===| |Track
'   |IR 2|---.-->| 2   7 |<-/\/\---|>|--{     '--^---|       | |A|---->
'   |RX 3|-. |  >| 3   6 |<-/\/\---|>|--{ 2k2     |/-'       | |C|---->
'   '----' '-|-->| 4   5 |<---------------/\/\-.--|  NFET     | | |Volts
'            |   '-------'   220  RdLed |      /  |\>-.       | |-|
'            |                           |   47k\      |       |  |
'            '---------------------------^------^------^-------^--'
'
'*****************************************************************
#picaxe 08m                         'Compile with 08m
'    Variables
symbol    SpdFlag   = b0           'Stop/Slow/Full Flag (0,1,2)
symbol    Loop1     = w1           'Loop Variable
'    Pins
symbol    HeadLt    = 0            'HWh LED on pin 0
symbol    BrakeLED  = 1            'Red LED on pin 1
symbol    FET       = 2            'PWM FET on pin 2
symbol    IRReceive = Pin3         'IR Rx   on pin 3
'    Constants
```

```
symbol    Version   = 4           'Version Number
symbol    StopVal   = 150         'Stop Value
symbol    SloSpeed  = 700         'Low Speed Value.
symbol    TopSpeed  = 1023        'Full Speed Value

'****************************************************************************
Init:                             'Initialize
     pause 500                    'Time for Startup
     high HeadLt                  'Turn On Headlight
     for loop1 = 1 to Version     'Flash LED Version times
          high BrakeLED           'Turn On Brake light
          pause 150               'Pause 200ms
          low BrakeLED            'Turn Off Brake light
          pause 150               'Pause 200ms
     next loop1                   'Again
     SpdFlag = 0                  'Set Flag to Stopped
     gosub Slow2Full              'Get up to Full

'****************************************************************************
RxIR:
     infrain2                     'Wait for IR Signal
     if infra = 1 then StopJ      'Stop
     if infra = 2 then SlowJ      'Slow Speed
     if infra = 3 then FastJ      'High Speed
     goto RxIR                    'Repeat forever

'****************************************************************************
'         **** Stop ****
'****************************************************************************
StopJ:
     if SpdFlag = 0 then RxIR     'Already Stopped
     if SpdFlag = 1 then DecSpd   'At Slow
     gosub Full2Slow              'Full to Slow
DecSpd:
     gosub Slow2Stop              'Slow to Stop
     low HeadLt                   'Turn Off Headlight
     goto RxIR                    'Next Command

'****************************************************************************
'         **** Slow Speed ****
'****************************************************************************
SlowJ:
     high HeadLt                  'Turn On Headlight
     if SpdFlag = 0 then IncSpd   'Need to Speed Up
     if SpdFlag = 1 then RxIR     'Already at Slow
     gosub Full2Slow              'Slow Down
     goto RxIR
IncSpd:
     gosub Stop2Slow              'Speed Up
     goto RxIR                    'Next Command

'****************************************************************************
'         **** Full Speed ****
'****************************************************************************
FastJ:
     high HeadLt                  'Turn On Headlight
     if SpdFlag = 2 then RxIR     'Already at Full
     if SpdFlag = 1 then Faster   'At Slow
     gosub Stop2Slow              'Stop to Slow
Faster:
     gosub Slow2Full              'Slow to Full
```

```
     goto RxIR                    'Next Command

'*************************************************************************
'         **** Subroutines ****
'*************************************************************************
Slow2Stop:                        'Slow To Stop
     high BrakeLED                'Turn On Brake light
     for Loop1 = SloSpeed to StopVal step -20   'Slow to Stop
          pwmout FET,255,Loop1    'PWM Output to FET
          pause 100               'Deceleration pause
     next Loop1                   'Slower
     pwmout FET,0,0               'PWM Off
     low BrakeLED                 'Turn Off Brake light
     SpdFlag = 0                  'Set Flag to Stop
     return                       'Done

'*************************************************************************
Full2Slow:                        'Option 2
     high BrakeLED                'Turn On Brake light
     for Loop1 = TopSpeed to SloSpeed step -20  'Slow Down
          pwmout FET,255,Loop1    'PWM Output to FET
          pause 100               'Deceleration pause
     next Loop1                   'Slower
     low BrakeLED                 'Turn Off Brake light
     SpdFlag = 1                  'Set Flag to Slow
     return                       'Done

'*************************************************************************
Stop2Slow:                        'Option 3
     for Loop1 = StopVal to SloSpeed step 20    'Speed Up
          pwmout FET,255,Loop1    'PWM Output to FET
          pause 100               'Acceleration pause
     next Loop1                   'Faster
     SpdFlag = 1                  'Set Flag to Slow
     return                       'Done

'*************************************************************************
Slow2Full:                        'Option 4
     for Loop1 = SloSpeed to TopSpeed step 20   'Speed Up
          pwmout FET,255,Loop1    'PWM Output to FET
          pause 100               'Acceleration pause
     next Loop1                   'Faster
     SpdFlag = 2                  'Set Flag to Fast
     return                       'Done




'*****************************************************************
'              IR Remote Control for Jigger - PICAXE 08M
'
'         46 bytes used of 256  -  7 October 2006  -  Version 3
'*****************************************************************
symbol     YellowLED  = 4            'Yellow LED on Output 4
symbol     SW_3       = pin3         'Switch 3
symbol     SW_2       = pin2         'Switch 2
symbol     IRLED      = 1            'IR LED on Output 1
symbol     SW_0       = pin0         'Switch 1
symbol     DataVar    = b0           'Data Variable
symbol     LoopVar    = b1           'Loop Variable

'*****************************************************************
```

```
MAIN:                                   'Loop until switch is pressed
     if SW_0 = 1 then TX_1              'Is switch 1 pressed
     if SW_2 = 1 then TX_2              'Is switch 2 pressed
     if SW_3 = 1 then TX_3              'Is switch 3 pressed
     goto main                          'Look again

TX_1:
     let DataVar = 1                    'Code = 1
     goto TX_IR
TX_2:
     let DataVar = 2                    'Code = 2
     goto TX_IR
TX_3:
     let DataVar = 3                    'Code = 3
     goto TX_IR

          'Transmit code 10 times for increased reliability
TX_IR:
     high YellowLED                     'Yellow LED on
     for LoopVar = 1 to 10              'Send infrared code 10 times
          Infraout IRLed, DataVar       'Send code
          pause 45                      'Wait 45 milliseconds
     next LoopVar                       'Again
     low YellowLED                      'LED off
     goto MAIN

'********************************************************************
```

Flanders waits for a green light to take the spare sleepers to the gang out on the main.

Go to

http://blod.dyndns.org/Picaxe/picaxeprojects.htm - **for my Picaxe Projects**

and

http://blod.dyndns.org/home/grailway/GRail.php - **for my Garden Railway**