

Here is the code for the :-

'2 Loop Train Controller

```
*****
'
'                MAIN Controller
*****
' 422 Bytes of 2048   -      18 January 2009
' Tested and OK     -      13 June 2009
'
*****
#picaxe 18x          'Compile with 18x
#gosubs 255         'Allow 255 Gosubs

'      8 Outputs
symbol   StnLts      = 0      'Station Lights
symbol   Control     = 1      'Start Train
symbol   Signal      = 2      'Signal Straight Output 1=Green 0=Red
symbol   PWMPin      = 3      'PWM Output
symbol   TAMotor     = 4      'TA Points Motor Control
symbol   TADir       = 5      'TA Points Branch Motor Direction
symbol   TBDir       = 6      'TB Points Branch Motor Direction
symbol   TBMotor     = 7      'TB Points Motor Control

'      5 Inputs
symbol   Mode        = pin0   'Mode Switch. 1=2 Trains 0=1 Train
symbol   ADCStn      = 1      'Station Wait ADC
symbol   ADCSpd      = 2      'Speed Max ADC
symbol   TrainDet    = pin6   '0 = Train detected approaching points
symbol   StnDet      = pin7   '0 = Train in Loop - Stn Lights on

'      Flags
symbol   Flag        = b0     'Flag Byte
symbol   FlagA       = bit0   '0 = Train to Stn & 1 = Train to Loop
symbol   FlagB       = bit1   '0 = 1 Train & 1 = 2 Trains

'      Variables
symbol   WaitTime    = b1     'Station wait time
symbol   Delay1      = b2     'Delay
symbol   Counter     = b3     'Variable Counter
symbol   Byte1       = b4     'Message Byte 1
symbol   Byte2       = b5     'Message Byte 2
symbol   Byte3       = b6     'Message Byte 3
symbol   SpeedVal    = b7     'Speed ADC
symbol   SpeedL      = b8     'Low Speed Value
symbol   SpeedH      = b9     'Hi Speed Value
symbol   SpeedCalc   = W5     'Calc for LCD
symbol   Speed       = W6     'PWM Speed Value

'      Constants
symbol   MotorOn     = 20000  'Motor On time * 10uS = 200ms
symbol   RelaySet    = 50     'Time for Relay to operate (mS)

*****
Init:
    gosub TALoop          'Points to Loop
    gosub TBLoop         'Points to Loop
    low Signal           'Set Signal Red
    low PWMPin           'Turn off PWM
    low Control          'Set 08M PWM to off
```

```

Flag = 0          'Clear Flags
pause 3000       'Pause 3Sec
if Mode = 1 then
  FlagB = 1      '2 Trains
  low Control    'Set 08M PWM to off
  high StnLts   'Turn on Stn Lights
else
  FlagB = 0      '1 Train
  high Control   'Set 08M PWM to High Speed
  low StnLts    'Turn off Stn Lights
endif
setint %00000000,%01000000 'Interrupt when pin6 Low

'*****
' Main Loop
'*****
Main:
  readADC ADCSpd, SpeedVal      'Get Max speed 0>255
  if SpeedVal < SpeedL then AlterPWM '-3 diff
  if SpeedVal > SpeedH then AlterPWM '+3 diff
  goto Main                    'Forever

AlterPWM:
  Speed = SpeedVal *2 + 495     'Calc Speed = 490>1000
  PWMout PWMPin,254,Speed      'Set Max Speed
  SpeedH = SpeedVal +3 MAX 255 'Set Comparison
  SpeedL = SpeedVal -3 MIN 3   'Set Comparison
  if FlagB = 1 then            '2 Trains
    gosub MTTwo                '
  else                          '1 Train
    gosub MTOne                 '
  endif
  pause 100
  goto Main                    'Forever

'*****
' Interrupt Control Subroutines
'*****
Interrupt:
  if FlagB = 0 then MODESingle 'FlagB = 0 = Single Train

ModeDouble:                    'DOUBLE TRAIN MODE

  if FlagA = 0 then            'FlagA = 0 = To Station
    gosub MT2Stn              'Message - Train to Station
    low StnLts                'Turn Off Stn Lights
    gosub TASTn               'Set PointA to Station
    high Signal               'Set Signal Green
    pause 200                 'Wait 200mS
    high Control              'Signal 08m - Start Train
    gosub TBStn               'Set PointB to Station
    gosub Look4StnDet         'Look for Stn Approach
    low Signal                'Set Signal Red
    low Control               'Signal 08m - Stop Train
    pause 1000                'Wait 1000mS
    high StnLts              'Turn on Stn Lights
    FlagA = 1                 'Next Time Train to Loop
  else
    gosub TBLoop              'Set PointB to Loop

```

```

        gosub MT2Loop           'Message - Train to Loop
        pause 1000             'Pause 1000mS
        gosub TALoop           'Set PointA to Loop
        pause 2000             'Pause 2000mS
        FlagA = 0              'Next Time Train to Stn
    endif
    goto RstInt

```

```

ModeSingle:
    if FlagA = 0 then
        gosub TBStn           'Set PointB to Station
        gosub MT2Stn          'Message - Train to Station
        gosub Look4StnDet     'Look for Stn Approach
        high StnLts           'Turn on Stn Lights
        low Control            'Signal 08m - Stop Train
        pause 2000            'Wait for Slowdown
        gosub StationWaitTime 'Station Delay Countdown
        gosub TASTn           'Set PointA to Station
        high Signal            'Signal Green
        pause 200              'Wait 200mS
        high Control           'Signal 08m - Start Train
        pause 1000            'Wait 1000mS
        low StnLts             'Turn Off Station Lights
        pause 1000            'Wait 1000mS
        low Signal             'Set Signal Red
        FlagA = 1              'Next Time Train to Loop
    else
        gosub TBLoop           'Set PointB to Loop
        gosub MT2Loop          'Message - Train to Loop
        pause 1000             'Pause 1000mS
        gosub TALoop           'Set PointA to Loop
        pause 2000             'Pause 2000mS
        FlagA = 0              'Next Time Train to Stn
    endif

```

```

RstInt:
    gosub MClrBot             'Clear Bottom Message Line
    setint %00000000,%01000000 'Interrupt when Remote Sensor LOW
    return                    'Done

```

```

'*****
' Subroutines
'*****

```

```

TALoop:
    high TADir                'TA Direction Relay on
    pause RelaySet            'Relay action time
    pulsout TAMotor, MotorOn  'Pulse Point Motor
    low TADir                 'Drop Relay
    return                    'Done

```

```

'*****

```

```

TASTn:
    pulsout TAMotor, MotorOn  'Pulse Point Motor
    return                    'Done

```

```

'*****

```

```

TBLoop:
    high TBDir                'TB Direction Relay on
    pause RelaySet            'Relay action time
    pulsout TBMotor, MotorOn  'Pulse Point Motor

```

```

low TDir                                'Drop Relay
return                                  'Done

'*****
TBStn:
  pulsout TBMotor, MotorOn              'Pulse Point Motor
  return                                  'Done

'*****
StationWaitTime:
  readADC ADCStn, Delay1                'Raw Delay 0 - 255
  WaitTime = Delay1 / 10 + 5            '5 - 30 Seconds
  for Counter = WaitTime to 0 step -1
    gosub MStnStop                      'Time countdown
    pause 1000                          'Wait 1 second
  next Counter                          'Again
  return                                  'Done

'*****
Look4StnDet:
  do until StnDet = 0                   'If detected exit
  loop                                   'Again
  return                                  'Done

'*****
' LCD Message Subroutines
'*****

MTOne:                                  'Single Train Mode
  Byte1 = 0                             'Message Number
  goto MSpd                              'Calc Speed
TTwo:                                    'Double Train Mode
  Byte1 = 1                             'Message Number

MSpd:
  SpeedCalc = SpeedVal*10/25 MAX 99 MIN 5 'Calc Speed
  if SpeedCalc <10 then                  'If < 10
    Byte2 = 0                            'Leading Zero
    Byte3 = SpeedCalc                    '0-9
  else                                    '>10
    Byte2 = SpeedCalc /10                '10's Byte
    Byte3 = SpeedCalc //10               ' 1's Byte
  endif
  goto Message

MT2Loop:                                 'Train to Loop
  Byte1 = 2                              'Message Number
  Byte2 = 1                              'Value required
  Byte3 = 1                              'Value required
  goto Message

MT2Stn:                                  'Train to Station
  Byte1 = 3                              'Message Number
  Byte2 = 1                              'Value required
  Byte3 = 1                              'Value required
  goto Message

MStnStop:                                'Wait at Station XXs
  Byte1 = 4                              'Message Number
  if Counter <10 then                   'If < 10
    Byte2 = 0                            'Leading Zero

```

```

        Byte3 = Counter          '0-9
    else                          '>10
        Byte2 = Counter /10      '10's Byte
        Byte3 = Counter //10     ' 1's Byte
    endif
    goto Message

MClrTop:                          'Clear Top Line
    Byte1 = 5                    'Message Number
    Byte2 = 1                    'Value required
    Byte3 = 1                    'Value required
    goto Message

MClrBot:                          'Clear Bottom Line
    Byte1 = 6                    'Message Number
    Byte2 = 1                    'Value required
    Byte3 = 1                    'Value required
    goto Message

Message:
    sertxd("AB",Byte1, #Byte2, #Byte3)
    return

'*****
'Variable Speed PWM for 2 Loop Train
'*****
#picaxe 08m                        'Compile with 08M
'    137 Bytes of 256                13 June 2009
'*****
'    **** I/O Pins ****
symbol    StartStop    = 0          'Relay for PWM to Start track
symbol    Stopped      = pin1      '0 if No current (Not Currently Used)
symbol    PWMPin       = 2          'PWM pin on 08M
symbol    Control      = pin3      'Input from 18X
symbol    HiSpd        = 4          'Max Spd ADC

'    **** Variables ****
symbol    Flag0        = bit0       'Flag 1=Max 0=Slow
symbol    SpdMax       = b1         'Max Speed 0-255
symbol    SpdMin       = b2         'Min Speed Set Var
symbol    SpdVar       = b3         'PWM Loop Var
symbol    StepVal      = b4         'Loop Step Value
symbol    SpdMax1      = b5         'Calc Max Speed
symbol    SpdVar1      = w5         'PWM Loop Var * 4
symbol    MaxSpd       = w6         'Calculated Val

'*****
Init:
    SpdMin      = 50                'Min Speed Value
    Flag0       = 0                'Set Flag to 0
    pwmout PWMPin,254,600          'PWM Out Slow

'*****

Main:
    do                          'Check for Control Change
        if Control = 1 and Flag0 = 0 then gosub SpeedUp
        if Control = 0 and Flag0 = 1 then gosub SlowDwn
    loop
    goto Main

```



```

#picaxe 18X                               'Compile with PICAXE-18X
#gosubs 255                                'Allow 255 gosubs

'      *** Outputs ***
symbol    BackLt    = 0                    'LCD Backlight
symbol    LED       = 1                    'Indicator LED
symbol    RS        = 2                    '0 = Command 1 = Data Register
symbol    En        = 3                    '0 = Idle    1 = Enable
symbol    DB4       = 4                    'LCD Data Line 4
symbol    DB5       = 5                    'LCD Data Line 5
symbol    DB6       = 6                    'LCD Data Line 6
symbol    DB7       = 7                    'LCD Data Line 7

'      *** Inputs ***
symbol    In0       = 0                    'Input Pin 0 - Not Used
symbol    In1       = 1                    'Input Pin 1 - Not Used
symbol    In2       = 2                    'Input Pin 2 - Not Used
symbol    In7       = 6                    'Input Pin 6 - Not Used
symbol    SIPin     = 7                    'Input Pin 7 - Serial In

'      *** Variables ***
symbol    IB1       = b1                    '1st Byte - Message Number
symbol    IB2       = b2                    '2nd Byte - Additonal Data
symbol    IB3       = b3                    '3rd Byte - Additonal Data
symbol    Counter   = b4                    'Loop Counter
symbol    CStart    = b5                    'Start EEprom location
symbol    CEnd      = b6                    'Stop  EEprom location
symbol    DByte     = b7                    'Common Data Byte
symbol    RSbit     = b8                    'Data/Command Mask

'      *** Constants ***
symbol    RSCMDmask = %00000000           'Select LCD Command register
symbol    RSDATmask = %00000100           'Select LCD Data register
symbol    SSPD      = N2400                'Serial Boardrate 4800 @ 8mHz

'*****
'      Load Permanent EEprom values
'*****
'      01234567890123456789
'      eeprom  0, (" Auto Train Control ")  'Init Message Top Line
'      eeprom 20, ("Murray Bold  June 09")  'Init Message Bot Line
'      eeprom 40, ("One Train  Spd =  %")   'Mess1 Top Line
'      eeprom 60, ("Two Trains Spd =  %")   'Mess2 Top Line
'      eeprom 80, (" Train to Loop  ")     'Mess3 Bot Line
'      eeprom 100, (" Train to Station ")   'Mess4 Bot Line
'      eeprom 120, ("Wait @ Station =  s")  'Mess5 Bot Line
'      eeprom 140, (" ")                   'Mess6 Clear Line
'      eeprom 250, ($33,$32,$28,$0C,$06,$01) 'LCD Init Bytes

'*****
'      Main code starts here
'*****
PowerOnReset:                                'Initialize LCD
  low BackLt                                'Turn Off Backlight Control
  for Counter = 250 to 255                   'Read 6 Bytes
    read Counter,DByte                       'Read EEPROM
    gosub SendInitCmdByte                    'Send Command Byte
  next                                       'Next Byte
  pause 500                                  'Let LCD Stabilize
  high BackLt                                'Turn On Backlight

```

```

DByte = 128:CStart= 0:CEnd= 19      'Output Line 1 'Program Name'
gosub DispLine                       'Send Command and Data
DByte = 192:CStart= 20:CEnd= 39     'Output Line 2 'Name & Date'
gosub DispLine                       'Send Command and Data
pause 3000                           'Time to read Startup Screen
setfreq m8                           'Receive at 4800bd

'*****
'          Read in Serial Data - "AB" Start Qualifier + 3 Data Bytes
'*****
ReceiveDataStart:
  serin SIPin,SSPD,("AB"),IB1,IB2,IB3 'Wait for Start Byte
  on IB1 gosub MTOne,MTTwo,MT2Loop,MT2Stn,MStnStop,MClrTop,MClrBot
  pulsout LED,20000                   '100mS Pulse @8mHz
  goto ReceiveDataStart

MTOne:          '1 Train Speed = xx%'
  DByte = 128:CStart= 40:CEnd= 59     'Output Line 1
  gosub DispLine                       'Send Command and Data
  gosub DispDataTop                   'Display Speed
  return

MTTwo:          '2 Trains Speed = xx%'
  DByte = 128:CStart= 60:CEnd= 79     'Output Line 1
  gosub DispLine                       'Send Command and Data
  gosub DispDataTop                   'Display Speed
  return

MT2Loop:        '  Train to Loop      '
  DByte = 192:CStart= 80:CEnd= 99     'Output Line 2
  gosub DispLine                       'Send Command and Data
  return

MT2Stn:         '  Train to Station   '
  DByte = 192:CStart=100:CEnd=119     'Output Line 2
  gosub DispLine                       'Send Command and Data
  return

MStnStop:       'Wait at Station xxS'
  DByte = 192:CStart=120:CEnd=139     'Output Line 2
  gosub DispLine                       'Send Command and Data
  gosub DispDataBot                   'Display Time Remaining
  return

MClrTop:        '          '          'Clear Top Line
  DByte = 128:CStart=140:CEnd=159     'Output Line 2
  gosub DispLine                       'Send Command and Data
  return

MClrBot:        '          '          'Clear Bottom Line
  DByte = 192:CStart=140:CEnd=159     'Output Line 2
  gosub DispLine                       'Send Command and Data
  return

'*****
'          Send Line of Data to LCD Registers from eeprom
'*****
DispLine:
  gosub SendCmdByte                   'Send Command
  for Counter = CStart to CEnd       'Bytes Start to End

```

```

        read Counter,DByte          'Read EEprom
        gosub SendDataByte          'Send Data
next                                     'Next Character
return

DispDataTop:
DByte = 145:gosub SendCmdByte        'Send Command
if IB2 = 0 then                       'If Zero Then Space
    DByte =" "                        'Insert Space
else
    DByte = IB2                       'Leave Same as not Zero
endif
gosub SendDataByte                    'Send Data
DByte = 146:gosub SendCmdByte        'Send Command
DByte = IB3:gosub SendDataByte      'Send Data
return

DispDataBot:
DByte = 209:gosub SendCmdByte        'Send Command
if IB2 = 0 then                       'If Zero Then Space
    DByte =" "                        'Insert Space
else
    DByte = IB2                       'Leave Same as not Zero
endif
gosub SendDataByte                    'Send Data
DByte = 210:gosub SendCmdByte        'Send Command
DByte = IB3:gosub SendDataByte      'Send Data
return

'*****
'           Write Data to LCD Command or Data registers
'*****
SendInitCmdByte:
    pause 30                          'Delay 15mS at 8MHz

SendCmdByte:
    peek $30,rsbit                    'Recover Out0 and Out1
    RSbit = RSbit & %11               'Send to Command register
    goto SendCmdOrDataByte

SendDataByte:
    peek $30,RSbit                    'Recover Out0 and Out1
    RSbit = RSbit & %11 | RSDATmask  'Send to Data register

SendCmdOrDataByte:
    pins = DByte & %11110000 | RSbit  'Put MSB out first
    pulsout En,2                       'Give a 10uS pulse on Enable
    pins = DByte * %00010000 | RSbit  'Put LSB out second
    pulsout En,2                       'Give a 10uS pulse on Enable
return

```

```

'*****

```