Model Railroad System

2.2.1

Generated by Doxygen 1.8.17

Chapter 1

Example Yard CTC Panels

The Chubb_FoxYard.tcl file contains part of the layout shown in Chapters 12 (Figure 12-11 on page 12-27) and 14 (Figure 14-3 on page 14-7) of The Computer/Model Railroad Interface User's Manual Version 3.

First we will connect to the Chubb CmriNet via a USB serial port (a USB <=> RS485 adaptor).

```
## Connect to the Chubb CmriNet
CmriSupport::CmriNode openport /dev/ttyUSB0
```

Then we will initialize the two SMINI nodes

```
# Two SMINI nodes -- see Chapter 14 of The Computer / Model Railroad Interface
# User's Manual for details -- note: this is only a subset of Figure 14-3.
CmriSupport::CmriNode EvertJct -type SMINI -address 3
CmriSupport::CmriNode Donaldson -type SMINI -address 2
```

Then we will create abstract objects for all trackwork.

```
## Create all abstract objects for all trackwork.
C4TSMINI_Switch sm10 -nodeobj Donaldson -motorport 4 -motorbit 6 \
      -plate TG10Plate -ossensorport 0 -osbit 3
TwoHead3over2 SIG14LAB -nodeobj Donaldson -port 3 -bit 0 -signalname SIG14LAB
C4TSMINI_Block BK15 -nodeobj Donaldson -port 0 -bit 4
OneHead3Color SIG16LA -nodeobj Donaldson -port 4 -bit 0 -signalname SIG16LA
OneHead3Color SIG16RA -nodeobj Donaldson -port 4 -bit 3 -signalname SIG16RA
C4TSMINI_Block BK16 -nodeobj EvertJct -port 0 -bit 0
OneHead3Color SIG18LA -nodeobj EvertJct -port 0 -bit 0 -signalname SIG18LA
OneHead3Color SIG18RA -nodeobj EvertJct -port 0 -bit 3 -signalname SIG18RA
C4TSMINI_Block BK17 -nodeobj EvertJct -port 0 -bit 1
ThreeHead3over2over2 SIG20RABC EvertJct -port 1 -bit 0 -signalname SIG20RABC
C4TSMINI_Switch sm11 -nodeobj EvertJct -motorport 0 -motorbit 6 \
      -plate TG11Plate -ossensorport 0 -osbit 2
OneHead3Color SIG20LB -nodeobj EvertJct -port 2 -bit 3 -signalname SIG20LB
OneHead3Color SIG20LD -nodeobj EvertJct -port 3 -bit 0 -signalname SIG20LD
OneHead3Color SIG20LA -nodeobj EvertJct -port 2 -bit 0 -signalname SIG20LA
C4TSMINI_ManualSwitch TF12 -nodeobj EvertJct -pointsenseport 0 \
      -pointsensebit 4 -ossensorport 0 -osbit 2
C4TSMINI_Block BK18 -nodeobj EvertJct -port 0 -bit 3
```

Then we will initialize the Direction Of Travel to no direction.

```
set DOT3 nodirection
```

Then in the main loop we will Invoke all trackwork and get occupicency.

```
# Main Loop Start
while {true} {
  # Invoke all trackwork and get occupicency
  MainWindow ctcpanel invoke FOXYard-1
  MainWindow ctcpanel invoke TG10
  MainWindow ctcpanel invoke FOXYard-2
  MainWindow ctcpanel invoke BK18-1
  MainWindow ctcpanel invoke TG11
  MainWindow ctcpanel invoke TF12
  MainWindow ctcpanel invoke BK18-2
```

```
MainWindow ctcpanel invoke BK18-3
MainWindow ctcpanel invoke BK18-4
MainWindow ctcpanel invoke BK18-5
MainWindow ctcpanel invoke BK18-6
MainWindow ctcpanel invoke BK15
MainWindow ctcpanel invoke BK16
MainWindow ctcpanel invoke FOXYard
MainWindow ctcpanel invoke BK17
```

And then activate the switch motors.

```
# Activate switch motors
MainWindow ctcpanel invoke TG10Plate
MainWindow ctcpanel invoke TG11Plate
```

We will then compute the direction of travel.

```
if {[sm10 occupied] && $DOT3 ne "Westbound"} {
    set DOT3 Eastbound
} elseif {[sm10 occupied] && $DOT3 ne "Eastbound"} {
    set DOT3 Westbound
}
if {![sm10 occupiedp] &&
    ![BK15 occupiedp] &&
    ![BK16 occupiedp] &&
    ![BK17 occupiedp] &&
    ![sm11 occupiedp]} {
    set DOT3 nodirection
}
```

We will then set the aspects of the Eastbound signals from west to east.

```
## Calculate SIG20RABC
SIG20RABC setaspect {red red red}
if {![sm11 occupiedp]} {
    if {[sm11 pointsense] eq "normal" && [TF12 pointsense] eq "reverse"} {
        SIG20RABC setaspect {red red yellow}
    } elseif {![BK18 occupiedp]} {
        if {[sm11 pointsense] eq "normal"} {
            SIG20RABC setaspect {yellow red red}
        } else {
            SIG20RABC setaspect {red yellow red}
        }
    }
}

## Calculate SIG18RA
SIG18RA setaspect red
if {$DOT3 ne "Westbound"} {
    if {![BK17 occupiedp]} {
        SIG18RA setaspect yellow
        if {[join [MainWindow ctcpanel getv SIG20RABC] -] ne "red-red-red"} {
            SIG18RA setaspect green
        }
    }
}
## Calculate SIG16RA
SIG16RA setaspect red
if {$DOT3 ne "Westbound"} {
    if {![BK16 occupiedp]} {
        SIG16RA setaspect yellow
        if {[MainWindow ctcpanel getv SIG18RA] ne "red"} {
            SIG16RA setaspect green
        }
    }
}
```

Then the aspects of the Westbound signals east to west.

```
## Calculate SIG14LAB
SIG14LAB setaspect {red red}
if {![sm10 occupiedp]} {
    if {[sm10 pointsense] ne "reverse"} {
        SIG14LAB setaspect {green red}
    } else {
        SIG14LAB setaspect {red yellow}
    }
}
## Calculate SIG16LA
```

```
  SIG16LA setaspect red
  if {$DOT3 ne "Eastbound"} {
      if {![BK15 occupiedp]} {
          SIG16LA setaspect yellow
      }
      if {[join [MainWindow ctcpanel getv SIG14LAB] -] ne "red-red"} {
          SIG16LA setaspect green
      }
  }
  ## Calculate SIG18LA
  SIG18LA setaspect red
  if {$DOT3 ne "Eastbound"} {
      if {![BK16 occupiedp]} {
          SIG18LA setaspect yellow
      }
      if {[MainWindow ctcpanel getv SIG16LA] ne "red"} {
          SIG18LA setaspect green
      }
  }
  ## Calculate SIG20LA, SIG20LB, SIG20LD
  SIG20LA setaspect red
  SIG20LB setaspect red
  SIG20LD setaspect red
  if {$DOT3 ne "Eastbound"} {
      if {![sm11 occupiedp]} {
          if {![BK17 occupiedp]} {
              set SIG20L yellow
              if {[MainWindow ctcpanel getv SIG18LA] eq "red"} {
                  set SIG20L green
              }
              if {[sm11 pointsense] eq "reverse"} {
                  SIG20LB setaspect $SIG20L
              } else {
                  if {[TF12 pointsense] eq "reverse"} {
                      SIG20LD setaspect $SIG20L
                  } else {
                      SIG20LA setaspect $SIG20L
                  }
              }
          }
      }
  }
```

Then we light the Direction Of Travel lamps

```
  MainWindow ctcpanel setv DOT3-West off
  MainWindow ctcpanel setv DOT3-East off
  if {$DOT3 eq "Eastbound"} {
      MainWindow ctcpanel setv DOT3-East on
  }
  if {$DOT3 eq "Westbound"} {
      MainWindow ctcpanel setv DOT3-West on
  }
```

Finally, the main loop ends.

```
  update;# Update display
}
```