APPENDIX J - DEFAULT RECEIVER PROGRAM

```
'
' PROGRAM: RXSLAVE.BS2
' Program run by User Programmable CPU in '97 FIRST Receiver Board
' Written by: Eric Rasmussen
' Date: 12/12/96
'
' Declare variables
'
x1          VAR byte
y1          VAR byte
x2          VAR byte
y2          VAR byte
aux1        VAR byte
wheel1      VAR byte
aux2        VAR byte
wheel2      VAR byte
tx_sw       VAR word
rx_sw       VAR word
sensor1     VAR byte
sensor2     VAR byte
relays      VAR word
'
' Define Aliases (variables which are sub-divisions of those defined above)
'                (aliases don't require any additional RAM)
'
sw1_fwd     VAR tx_sw.bit0 ' Aliases for each TX switch input
sw1_rev     VAR tx_sw.bit1
sw2_fwd     VAR tx_sw.bit2
sw2_rev     VAR tx_sw.bit3
sw3_fwd     VAR tx_sw.bit4
sw3_rev     VAR tx_sw.bit5
sw4_fwd     VAR tx_sw.bit6
sw4_rev     VAR tx_sw.bit7
sw5_fwd     VAR tx_sw.bit8
sw5_rev     VAR tx_sw.bit9
sw6_fwd     VAR tx_sw.bit10
sw6_rev     VAR tx_sw.bit11
sw7_fwd     VAR tx_sw.bit12
sw7_rev     VAR tx_sw.bit13
sw8_fwd     VAR tx_sw.bit14
sw8_rev     VAR tx_sw.bit15
rx_sw1      VAR rx_sw.bit0      ' Aliases for each RX switch input
rx_sw2      VAR rx_sw.bit1
rx_sw3      VAR rx_sw.bit2
rx_sw4      VAR rx_sw.bit3
rx_sw5      VAR rx_sw.bit4
rx_sw6      VAR rx_sw.bit5
rx_sw7      VAR rx_sw.bit6
rx_sw8      VAR rx_sw.bit7
rx_sw9      VAR rx_sw.bit8
rx_sw10     VAR rx_sw.bit9
rx_sw11     VAR rx_sw.bit10
rx_sw12     VAR rx_sw.bit11
rx_sw13     VAR rx_sw.bit12
rx_sw14     VAR rx_sw.bit13
```

```
rx_sw15   VAR rx_sw.bit14
rx_sw16   VAR rx_sw.bit15
rly1_fwd  VAR relays.bit0      ' Aliases for each relay output
rly1_rev  VAR relays.bit1
rly2_fwd  VAR relays.bit2
rly2_rev  VAR relays.bit3
rly3_fwd  VAR relays.bit4
rly3_rev  VAR relays.bit5
rly4_fwd  VAR relays.bit6
rly4_rev  VAR relays.bit7
rly5_fwd  VAR relays.bit8
rly5_rev  VAR relays.bit9
rly6_fwd  VAR relays.bit10
rly6_rev  VAR relays.bit11
rly7_fwd  VAR relays.bit12
rly7_rev  VAR relays.bit13
rly8_fwd  VAR relays.bit14
rly8_rev  VAR relays.bit15
'
' Define Constants
'
MASTERCPU CON 0       ' Pin used to communicate with Master CPU
FPIN      CON 1       ' Pin used to for flow control with Master CPU
RLYDATA   CON 5       ' Shift Register Data Pin for Relay Outputs
RLYCLOCK  CON 6       ' Shift Register Clock Pin for Relay Outputs
RLYLATCH  CON 7       ' Shift Register Latch Pin for Relay Outputs
SWDATA    CON 8       ' Shift Register Data Pin for Switch Inputs
SWCLOCK   CON 9       ' Shift Register Clock Pin for Switch Inputs
SWLATCH   CON 10      ' Shift Register Latch Pin for Switch Inputs
SSC       CON 11      ' Pin used to communicate with Serial Servo Controller
ADCDATA   CON 12      ' Analog to Digital Converter Data Pin
ADCCLOCK  CON 13      ' Analog to Digital Converter Clock Pin (for data)
ADC1SEL   CON 14      ' Pin to select ADC #1 (selected = low)
ADC2SEL   CON 15      ' Pin to select ADC #2 (selected = low)
GOODLED   CON 3       ' Pin used to turn status LED green
ERRLED    CON 4       ' Pin used to turn status LED red
SSCBAUD   CON $0020   ' Baud rate for communications with SSC
BS2BAUD   CON $4020   ' Baud rate for communications with BS2 (Master CPU)
PWM1      CON 8       ' Define SSC addresses for PWM outputs
PWM2      CON 9
PWM3      CON 10
PWM4      CON 11
PWM5      CON 12
PWM6      CON 13
PWM7      CON 14
PWM8      CON 15
SSC_CMD   CON 255     ' Command Prefix for SSC
'
' Initialize Hardware and Variables on Power Up or Reset
'
' All pins default to low (ground) & input (floating) on powerup or reset.
'
Output SSC          ' Prevent floating before data is sent (low)
Output FPIN
Output RLYDATA
Output RLYCLOCK
Output RLYLATCH
Output SWDATA
Output SWCLOCK
Output SWLATCH
```

```
Output GOODLED
Output ERRLED
High SWLATCH          ' These should be normally high
High ADC1SEL
High ADC2SEL
' Set Relay Outputs all off (in case shift register contains junk)
Shiftout RLYDATA, RLYCLOCK, LSBFIRST,[0\16]
Pulsout RLYLATCH, 1
'
' Main Program
'
Loop:

  ' Get data from Master CPU
  Serin MASTERCPU\FPIN, BS2BAUD, [Wait(255,255), x1, y1, x2, y2,
tx_sw.lowbyte, aux1, wheel1, tx_sw.highbyte, aux2, wheel2]

  High GOODLED ' Turn LED Green to show loop is running
  Serout SSC, SSCBAUD, [SSC_CMD, PWM1, x1] ' Update PWM outputs
  Serout SSC, SSCBAUD, [SSC_CMD, PWM2, y1]
  Serout SSC, SSCBAUD, [SSC_CMD, PWM3, x2]
  Serout SSC, SSCBAUD, [SSC_CMD, PWM4, y2]
  Serout SSC, SSCBAUD, [SSC_CMD, PWM5, aux1]
  Serout SSC, SSCBAUD, [SSC_CMD, PWM6, wheel1]
  Serout SSC, SSCBAUD, [SSC_CMD, PWM7, aux2]
  Serout SSC, SSCBAUD, [SSC_CMD, PWM8, wheel2]
  Low GOODLED  ' Turn LED off

  ' Read Switch Inputs into rx_sw
  Gosub ReadSwitches

  ' Read Analog Sensor Inputs into sensor1 & sensor2
  Gosub ReadSensors

  ' Set relays to match TX switches
  relays = tx_sw

  ' Use 1st 8 RX switches (rx_sw1-8) as STOP switches for Relays 1-4
  relays.lowbyte = relays.lowbyte &~ rx_sw.lowbyte

  ' Use 2nd 8 RX switches (rx_sw9-16) as GO switches for Relays 5-8
  relays.highbyte = relays.highbyte | rx_sw.highbyte

  ' Set Relay Outputs
  Shiftout RLYDATA, RLYCLOCK, LSBFIRST,[relays\16]
  Pulsout RLYLATCH, 1

Goto Loop: ' Start over at Loop:
End ' It should actually never get here!
'
' Subroutines
'
ReadSwitches: ' Reads receiver switch inputs into variable rx_sw
  Pulsout SWLATCH, 1
  Low SWLATCH
  Pulsout SWCLOCK, 1
  High SWLATCH
  Shiftin SWDATA, SWCLOCK, LSBPRE, [rx_sw\16] ' Read in the data
  rx_sw = rx_sw ^ 65535 ' Invert bits so that a closed switch is true (1)
Return
```



```
ReadSensors: ' Read receiver sensor inputs into variables sensor1 & sensor2
  Low ADC1SEL                    ' Select the 1st Analog to Digital Converter
  Pulsout ADCCLOCK, 1
  Shiftin ADCDATA, ADCCLOCK, msbpost, [sensor1] ' Read in the data
  High ADC1SEL                                  ' Deselect ADC1
  Low ADC2SEL                    ' Select the 2nd Analog to Digital Converter
  Pulsout ADCCLOCK, 1
  Shiftin ADCDATA, ADCCLOCK, msbpost, [sensor2] ' Read in the data
  High ADC2SEL                                  ' Deselect ADC2
Return
```