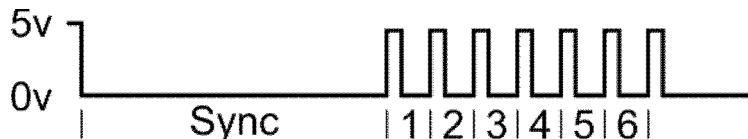


VEX Receiver Decoder

Circuit / Firmware by Jon Williams (jwilliams@efx-tek.com)

Theory of Operation

The output from the VEX receiver is an open collector PPM (pulse position modulation) stream that is common in the RC control world. The idle/spacing state of the output will be high via the R2 pull-up resistor. The PPM signal stream from the receiver into the processor looks like this:



A “cycle” begins with a low-going sync pulse that is about nine milliseconds in duration. At the end of the sync pulse the PPM pin will go high; this is the beginning of the Servo 1 timing. At this point the Servo 1 output is enabled and will stay on until the next low-to-high transition of the PPM pin; at this point the Servo 1 output is turned off and the Servo 2 output activated. Note that if you look at the PPM waveform on an oscilloscope only the low-going portion of each servo frame changes; the (high-going) framing pulse is constant (~500 uS); the low portion will vary from ~500 to ~1500 uS.

After four servos the firmware measures the low-going pulse of channels 5 and 6 to determine which button (top = short, bottom = long) was pressed for these channels. Pressing the top button on channels 5 or 6 activates the associated control output. Pressing the bottom button on channels 5 or 6 deactivates the output. The control outputs do not change state if neither or both buttons for a channel are pressed.

The firmware below was written for the Parallax (www.parallax.com) SX microcontroller using the [free] SX/B compiler. It is straightforward and easily translated to other processors/languages.

```
' =====
'
' File..... VEX_Demod-v2.SXB
' Purpose... Servo demodulator for VEX receiver
' Author.... Jon Williams
'           Copyright (c) 2007 Jon Williams
'           Some Rights Reserved
'           -- see http://creativecommons.org/licenses/by/2.5/
' E-mail.... jwilliams@efx-tek.com
' Started...
' Updated... 02 MAR 2007
'
' =====

'
' Program Description
' -----
'

' Simple program to convert PPM stream from VEX RC receiver to servo
' output pulses.

' See: http://www.vexfan.com/viewtopic.php?t=227

' Version 2: converts output of channels 5 and 6 to digital control; the
' upper button for these channels turns the output bit on, the lower
```

```

' button turns them off.

'
' ----- Conditional Compilation Symbols -----
'

' ${DEFINE Use_SX20_OFF}
' ${DEFINE Test_Mode_Off}

'
' ----- Device Settings -----
'

DEVICE      SX28, OSCXT2, TURBO, STACKX, OPTIONX, BOR42
FREQ        50_000_000
ID          "VexDemod"

'
' ----- IO Pins -----
'

PPM          PIN     RA.0 INPUT           ' PPM in (pull-up w/10k)
SvoPort      PIN     RB    OUTPUT
Servo1       PIN     RB.0
Servo2       PIN     RB.1
Servo3       PIN     RB.2
Servo4       PIN     RB.3
Control1     PIN     RB.4           ' digital control pins
Control2     PIN     RB.5

'
' ----- Constants -----
'

SvoMask      CON     %11110000         ' preserve RB.7..RB.4
IsOn         CON     1
IsOff        CON     0

'
' ----- Variables -----
'

pulseTmr     VAR     Word
svoPntr      VAR     Byte

tmpB1        VAR     Byte           ' work vars
tmpB2        VAR     Byte
tmpW1        VAR     Word

```

```

' =====
' PROGRAM Start
' =====

Start:
' {$IFDEF Use_SX20_OFF}
    PLP_C = %00000000                                ' pull up unused pins
' {$ENDIF}

    PLP_A = %0001
    SvoPort = %00000000

Wait_Forum_Sync:
    pulseTmr = 0
    DO WHILE PPM = 0
        PAUSEUS 1
        INC pulseTmr
    LOOP
    IF pulseTmr < 4000 THEN Wait_Forum_Sync

Main:
    svoPntr = %0000_0001
    DO
        SvoPort = SvoPort | svoPntr                  ' start servo pulse
        DO WHILE PPM = 1
            PAUSEUS 1                                ' complete framing pulse
        LOOP
        DO WHILE PPM = 0
            PAUSEUS 1                                ' complete timing pulse
        LOOP
        SvoPort = SvoPort & SvoMask
        svoPntr = svoPntr << 1                      ' clear servo outputs
    LOOP UNTIL svoPntr = %0001_0000                 ' point to next servo

Ctrl_Port1:
    DO WHILE PPM = 1                                ' complete framing pulse
    LOOP
    pulseTmr = 0
    DO WHILE PPM = 0
        PAUSEUS 5                                  ' measure timing pulse
        INC pulseTmr
    LOOP

    ' {$IFDEF Test_Mode}
    WATCH pulseTmr
    BREAK
    GOTO Start
    ' {$ENDIF}

    IF pulseTmr < 150 THEN                         ' short pulse?
        Control1 = IsOn
    ELSE
        IF pulseTmr > 250 THEN                     ' long pulse?
            Control1 = IsOff
        ENDIF
    ENDIF

Ctrl_Port2:
    DO WHILE PPM = 1

```

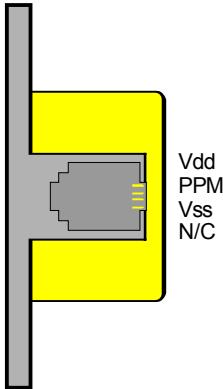
```

LOOP
pulseTmr = 0
DO WHILE PPM = 0
PAUSEUS 5
INC pulseTmr
LOOP
IF pulseTmr < 150 THEN
Control2 = IsOn
ELSE
IF pulseTmr > 250 THEN
Control2 = IsOff
ENDIF
ENDIF

GOTO Wait_For_Sync

```

VEX Receiver Connections

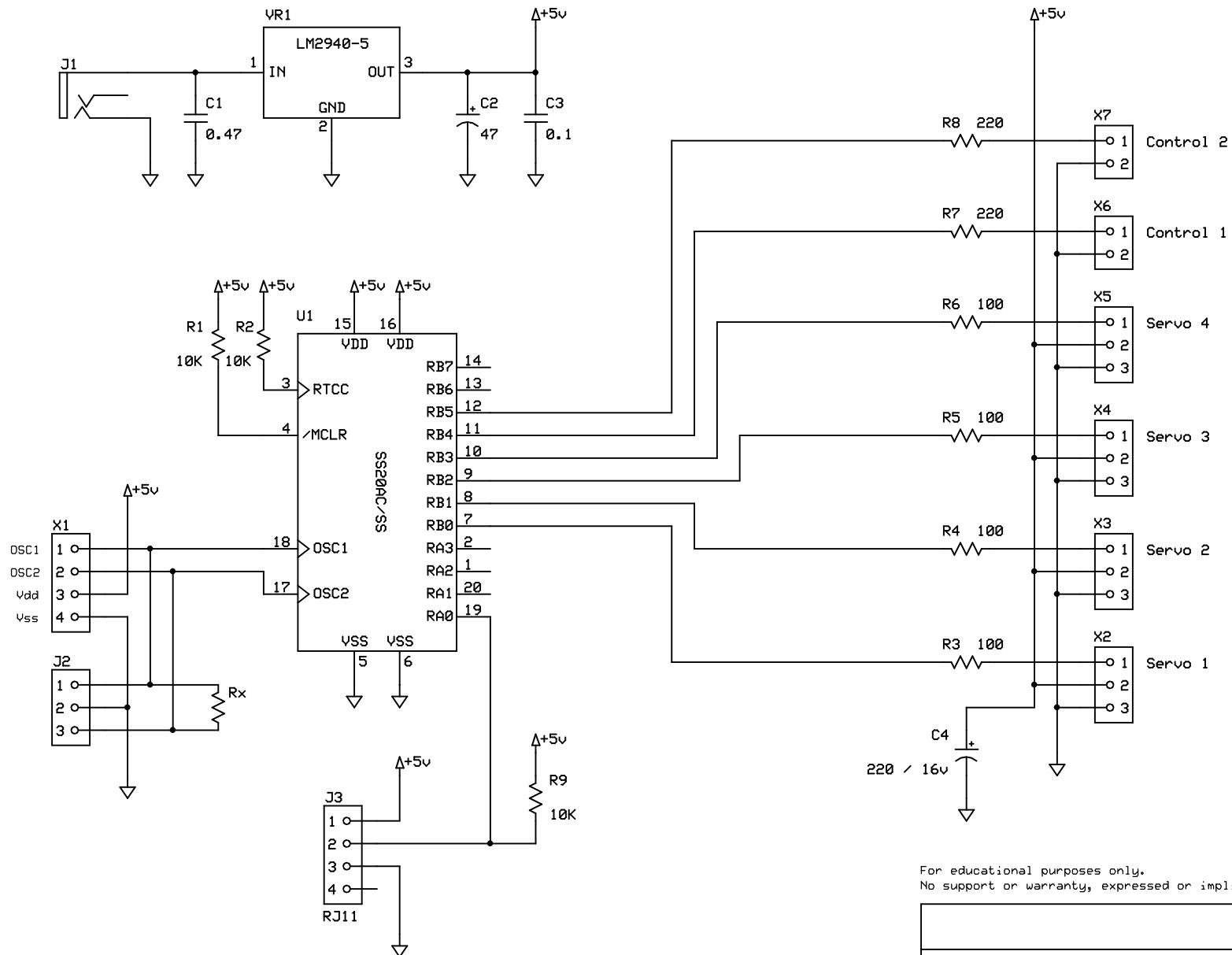


Terms of Use

This documentation is made available for educational purposes only. Neither Jon Williams, nor any host of this document, assume responsibility for completeness, accuracy, errors or omissions, or for the suitability of this circuit and code for any application. Connecting the VEX receiver to custom hardware may void VEX product warranties.

About the Author

In addition to his work with EFX-TEK, Jon Williams writes for Nuts & Volts magazine (“Stamp Applications” column) and is a free-lance electronics engineer who lives and works in the Los Angeles area. He can be reached at jwilliams@efx-tek.com.



For educational purposes only.
No support or warranty, expressed or implied.

VEX Receiver Decoder

Jon Williams

Rev 1.0
3/2/2007

Page # or name