synthiam.com

Controlling A Kangaroo With The V4

For Dave and Richard and anyone else who is interested, here is fully working and tested code for using the Kangaroo (with bi-directional comms) via the UART on the V4 with encoder feedback.
I am still quite new with EZ scripting so the code may not be as elegant as I would like, I could not find an ASCII to decimal function in EZ-Builder so I had to write a block of code to do this DJ maybe I am missing...

Last Updated: 9/17/2015

Step 1

For Dave and Richard and anyone else who is interested, here is fully working and tested code for using the Kangaroo (with bi-directional comms) via the UART on the V4 with encoder feedback.
I am still quite new with EZ scripting so the code may not be as elegant as I would like, I could not find an ASCII to decimal function in EZ-Builder so I had to write a block of code to do this DJ maybe I am missing something here?
Anyway I think this does show how fantastic and versatile the V4 is, well done DJ and team!
Tony

```

---

# Name : V4_Roo_serial

# Author : Tony Ellis

# Date : 01/07/2014

# Version : 1.0

---

definearray($byte,10)
$wheel_turn=63890
$dly=50
$FLAG=0
uartinit(0,0,19200)  # initalise UART

## ***** Initalise Kangaroo

uartwrite(0,0,"D,start",13)  # drive channel start
uartwrite(0,0,"T,start",13)  # turn channel start
uartwrite(0,0,"T,p0",13)  # this is also needed at start
#goto(START)
$val=2*$wheel_turn  # load position
$COMM_CHR="DF3"  # D=drive channel - F=move forward - 3=speed 3
goto(INC_POS)
halt()
:START
$val=$wheel_turn-500
$position=0
$unit=round($val/8)
$str="4000"
$x=1
goto(RAMP)
$str="7000"
```

```
$x=2
goto(RAMP)
$str="10000"
$x=3
goto(RAMP)
$str="7500"
$x=6
goto(RAMP)
$str="5500"
$x=7
goto(RAMP)
$str="2500"
$x=8
goto(RAMP)
uartwrite(0,0,"D,s0",13)
halt()
:RAMP
uartwrite(0,0,"D,s"+$str,13)
$v=$unit*$x
REPEATUNTIL($position>$v)
goto(GET_POS)
ENDREPEATUNTIL
print("position="+$position)
print("speed="+$str)
return
```

Subroutines

***** POSITION commands *****

```
:INC_POS                        # increment position subroutine
$str=tostring($val)               # convert variable to string
$COMM="pi"                        # determine direction
IF (getcharat($COMM_CHR,1)="R")
$COMM="pi-"
ENDIF
goto(SEND_DATA)
goto(GET_POS)
REPEATUNTIL($FLAG=1)
goto(GET_POS)
ENDREPEATUNTIL
print("POSITION = "+$position)
return
:GET_POS                        # get current position subroutine
uartwrite(0,0,"D,getp",13)
sleep($dly)
goto(GET_DATA)
print($position)
return
```

```
:SEND_DATA
$M_SPEED=(getbyteat($COMM_CHR,2)-48)*1200
$string=(getcharat($COMM_CHR,0)+","+$COMM+$str+"s"+$M_SPEED) # build data packet
uartwrite(0,0,$string,13)                                    # send packet
sleep($dly)                                                  # delay required
return
```

```
:GET_DATA
$num_bytes=uartavailable(0,0)
$data=uartread(0,0,$num_bytes)
$byte1=getbyteat($data,0)    # byte 1 = command
$byte2=getbyteat($data,1)    # byte 2 = ","
$byte3=getbyteat($data,2)    # byte 3 = command completed status
```

***** Convert ASCII digits to decimal number

```
$total=0
$multiplier=1
$x=$num_bytes-2
REPEATUNTIL($x=3)#$num_bytes-7)
$byte[$x]=getbyteat($data,($x-1))
$byte[$x]=($byte[$x]-48)*$multiplier*
*$total=$total+$byte[$x]*
*$multiplier=$multiplier*10
$x--
ENDREPEATUNTIL
```

***** Determine byte 3

$FLAG=0

```
IF ($byte3=80)
$FLAG=1      # "P" returned so move completed
ELSEIF ($byte3=112)
$FLAG=2      # lowercase "p" so move not completed
ELSEIF ($byte3=83)
$FLAG=3      # "S" returned so speed reached
ELSEIF ($byte3=115)
$FLAG=4      # lowercase "s" so still accelerating
ENDIF
IF ($FLAG<3)
$position=$total
ELSEIF ($FLAG>2)
$speed=$total
ENDIF

return
```