

Propeller Object Exchange

Contributed Code for the Propeller Microcontroller

[Log in/Register](#)



[All](#) [Data Storage](#) [Display](#) [Fun](#) [Human Input](#) [Math](#) [Motor Control](#) [Protocol](#) [Sensor](#) [Signals](#) [Speech/Sound](#) [Tool](#)

[Home](#) ›

Multi Propeller - 8bit Parallel IO

By: Mark Owen, created: 2014-12-07 | updated: 2014-12-09



Total votes: 0

[Request group membership](#)



Some simple mechanisms for transmission of arrays of data in eight bit parallel chunks. Sender/receiver synchronization is accomplished using a request to send (RTS) signal initiated by the sender. Upon sensing an RTS signal the receiver raises a clear to send signal (CTS) when it is ready to receive the data. The sender drops RTS once the data bits are set on the pins and waits for the receiver to drop CTS thereby indicating it has received the data. A checksum byte is calculated and transmitted following the last data byte. The calculated checksum is returned to the caller and may be examined by calling GetResult. In addition, upon completion of a receive the internally computed checksum can be examined by calling GetValue. The receiver can thereby detect a parity error by comparing GetResult with GetValue. If they disagree, a parity error has occurred.

The data are buffered in COG memory. Methods are provided to load the buffer from hub RAM prior to transmission, transmit the buffer, receive a buffer and to unload the buffer to hub RAM.

The COG buffer is 1KB in size. As a result transmissions may not exceed that size (1024 bytes or 256 longs). It can however be increased up to 1380 bytes (345 longs) if needed by changing the constant MSIZE. At MSIZE=345 all available cog memory is in use.

This object requires one cog for operation.

Clocked at ~390kBytes (3.12Mbits) per second on a pair of 80MHz Propeller systems sending 1,024 Byte blocks. If you need more speed you can a) over-clock the processors; b) modify the TxWait

function to reduce the pulse duration of the RTS and data signals
or c) reduce the PIO_waitinc value in Start to shorten the wait
time to less than its current ~476nS value.

Requires 10 I/O pins for operation:

RTS
CTS
MSB...LSB - eight contiguous pins

Example Usage Code:

```
CON
_clkmode = xtal1 + pll16x      ' System clock → 80 MHz
_xinfreq = 5_000_000          ' external crystal 5MHz

BUFFERSIZE  = PIO#MSIZE
PACKETSIZELONGS= 256
PACKETSIZEBYTES= PACKETSIZELONGS<<2

DATpin      = 0      ' 0..7 data bits
RTSpin      = 8      ' request to send pin
CTSpin      = 9      ' clear to send pin

OBJ
PIO : "ParallelIO"
VAR
long buffer[BUFFERSIZE]

PUB ATransmitter { run this on one propeller }
PIO.Start(PIO#AS_TRANSMITTER,DATpin, RTSpin, CTSpin)
repeat
PIO.LoadBuffer(@buffer,PACKETSIZELONGS)
if not PIO.TransmitBuffer(PACKETSIZEBYTES)
{ deal with transmit timeout }
else
{ do what you will with the buffer content }

PUB AReceiver { runthis on another propeller }
PIO.Start(PIO#AS_RECEIVER, DATpin, RTSpin, CTSpin)
repeat
if PIO.ReceiveBuffer(PACKETSIZEBYTES)
if PIO.GetResult <> PIO.GetValue ' checksum error
{ deal with the parity error }
else
PIO.UnloadBuffer(@buffer,PACKETSIZELONGS)
{ do what you will with the buffer content }
else
```

```
{ deal with receive timeout }
```

```
PUB AModeSwitcher
{to switch modes}
PIO.Start(<<true or false>>, DATpin, RTSpin, CTSpin)
..etc..
```

Items per page

1

Original File Upload

Attachment	Size
 ParallelIO.spin	67.24 KB

[Log in](#) or [register](#) to post comments

Home



Please contact obex.support@parallax.com with comments or questions.