Maximize Throughput while Polling with the HumPRO™ Series RF Transceiver

Reference Guide RG-00108

Introduction

Establishing a polling system within a network is a good way to ensure that each node has the ability to get its information across the link. This system establishes a star network where one module is the leader or Access Point (AP) that aggregates the data from all of the other nodes.

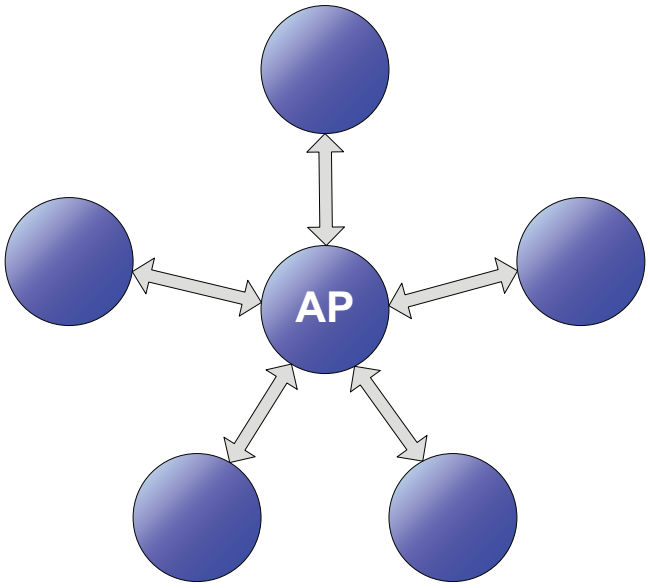


Figure 1: Star Network

The Access Point gets data by either sending a request message to specific nodes requesting their data or by sending a beacon message that the nodes use to time sending their data. A basic polling system is appropriate for systems with infrequent updates or small amounts of data. A beacon system is better for maximizing the data throughput.

This note describes a beacon system in detail, though a basic polling system is described in Example 3 later in the note. Since the HumPRO™ Series is a Frequency Hopping Spread Spectrum (FHSS) radio, the beacon serves to synchronize the nodes to the Access Point's frequency hopping pattern as well as serve as a baseline for timing the data transmissions.

Basic Concepts

There are some basic concepts and definitions that should be established before describing the system. An FHSS system works by transmitting on multiple frequencies across the allowed band. The radio stays on each channel for a certain time (the dwell time), and then hops to another frequency.

One of the features of this system is coordinating the transmitter and receiver jump to the same frequency at the same time. This requires good timing and synchronization between both sides. The HumPRO™ bases the timing on the transmitter and the receivers synchronize on the first packet that is transmitted on every hop. The transmitter sends an extended preamble on the first packet of each hop so that receivers can hop, pick up and lock on to the preamble so that data is not missed. This timing needs to be taken into account when setting up a polling system.

The polling system uses a beacon message sent by the Access Point on every hop to make sure that all of the nodes are synchronized to it. From there, the nodes can send their data either using a clock or timer or using the Carrier Sense Multiple Access (CSMA) algorithm that is built into the HumPRO™.

Synchronization and Timing

Maximizing the throughput of a polled data collection system can be achieved by synchronizing the polling with the channel hop timing inherent in the HumPRO™ Series. To synchronize, the beacon transmission interval must be at least the module hop interval. For the single hop scheduling described here, the best performance is an interval equal to the hop interval.

After an initial transmission, the transmitter and all nodes which receive the message stay on the same frequency channel for approximately 400 ms (390 ms for the high RF data rate) from the beginning of the transmission. At the end of that period, all nodes change to the next channel in the selected hop sequence. Transmission packets are not split across channels. It is possible for groups of bytes input to the module to be transmitted across channels. If the number of bytes to be sent is greater than what can go in the time left on the current hop, then the module sends what it can and sends the rest in a new packet on the next channel (if option TXPKT is off; factory default configuration).

There are two advantages to synchronizing to the channel hop timing:

1. Avoiding splitting messages across channels, which can result in getting part of a message if one of the two packets is not received
2. Avoiding the variation in timing due to the extended preamble that is required on the first transmission in a channel slot.

Figure 2 gives transmission timing for packet components based on data rate, address mode, encryption option, and number of payload bytes.

The RX verify time is from the PA_EN line going low at the end of packet on the transmitter until either the EX line is raised on the receiver (if EX_RXWAIT is set) or the start of data from the receiver UART (if packet mode is not used).

If CSMA is enabled, there is a variable delay before transmission, depending on channel activity. The table shows the minimum CSMA timing, if enabled.

The Sync Extension is applied on the first packet on a new channel or on any packet when bit 3 of ADDMODE (“Send long preamble”) is set.

Transmit Times

UART Baud Rate (bps)	RF Baud Rate (bps)	TX Setup Time	Min. CSMA	Extended Preamble	Address Mode			ACK Packet	Per Payload Byte	RX verify
					DSN	User	Extended User			
		Std / Enc			Std / Enc	Std / Enc	Std / Enc			Std / Enc
9,600 – 19,200	19,200	2.0 / 2.6+0.005*n	2.0	66.70	10.98 / 14.85	11.95 / 14.85	13.89 / 17.77	7.92 / 12.09	0.4863	1 / 1.0+0.01*n
38,400 – 115,200	152,340	2.0 / 2.8+0.005*n	2.0	24.65	1.54 / 2.03	1.73 / 2.09	1.91 / 2.27	1.15 / 1.67	0.06079	1.2 / 1.5+0.01*n

n = number of payload bytes

Figure 2: HumPRO™ Series Transceiver Transmit times (ms)

The number of payload bytes (user bytes) is multiplied by the Per Payload Byte time. There is an additional delay of up to one payload byte time based on the number of total bytes in the packet. For a maximum estimate, increase the payload count by 1.

For example, if UARTBAUD is 0x03 (38,400 bps), Extended User address mode is used with encryption (ADDMODE = 0x27), the payload is 15 bytes, and CSMA is disabled, the transmission time (not first on channel) is:

$$2.27 + 15 * 0.06079 = 3.18\text{ms}$$

If this is the first packet transmitted on the channel, add an additional 24.65ms.

In addition to the RF transmission time, the total time from start of first data byte to the transmitter UART until the last data byte is received from the receiver UART includes the following:

1. UART transmission time (10 bits per byte)
2. Trigger time if DATATO used to trigger transmission (DATATO-1 to DATATO ms)
3. Transmitter setup time (shown in table)
4. CSMA carrier acquisition, if CSMA is non-zero (dependent on external RF)
5. Receiver verification time for received packet (shown in table)
6. UART transmission time from receiver (10 bits per byte)

The simplest method is to poll each remote unit for data, then wait the maximum time for a response.

For more efficient processing, one beacon message can schedule multiple responses within the same channel slot time. Each node transmits at a specific time after the beacon message is received, maximizing the available channel time. If this method is used, a guard time should be used between scheduled responses to prevent missing messages due to jitter in response times. A guard time of approximately 2ms should be left at the end of the channel slot to ensure that the response will not be deferred until the next channel slot.

If one beacon message schedules multiple responses, a means should be provided to determine the identity of each response message. This can come from the message payload or the RXPKT mode can be used to read the message header, containing the source address, as well as the payload.

For examples, consider a network with one Access Point and four nodes. Extended User Addressing Mode is used and each node sends 100 bytes of data at the 115,200bps UART rate. First, establish the network addressing.

The UMASK register of all modules is set to 00 00 00 FF (hex). This establishes a network of up to 255 nodes.

Module 0 is set as the Access Point. It is given the USRCID = 76 54 32 00. Its UDESTID is set to 76 54 32 FF, which is the network broadcast address.

The rest of the modules get sequential USRCID addresses. They all have UDESTID = 76 54 32 00 so that they send their data to the Access Point.

Module Address Configurations

	Module 0	Module 1	Module 2	Module 3	Module 4
UMASK	00 00 00 FF	00 00 00 FF	00 00 00 FF	00 00 00 FF	00 00 00 FF
USRCID	76 54 32 00	76 54 32 01	76 54 32 02	76 54 32 03	76 54 32 04
UDESTID	76 54 32 FF	76 54 32 00	76 54 32 00	76 54 32 00	76 54 32 00

Figure 3: Module Address Configurations (hex values)

Acknowledgements should not be used for a polling system.

Example 1

In this example, each node gets a time slot to transmit based on their address. The timing begins from when the modules output the first byte of the beacon message on their UART. From that point each module waits a certain amount of time before transmitting. This is outlined in Figure 4.

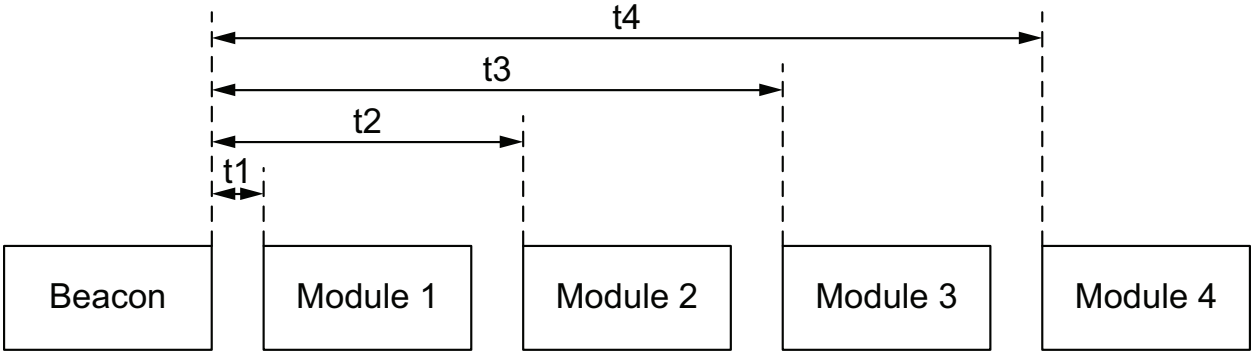


Figure 4: Module Transmission Timing

Since each module has sequential addresses a wait time is calculated and then multiplied by the address. The detailed timing values that need to be considered are shown in Figure 5.

The processor wait times are application specific. The rest of the values are given in Figure 2. It is worth noting that the UART In and TX Setup times can overlap with the Access Point Processor time. The amount of overlap is also application specific but can easily be calculated and tested.

Here is an overview of the assumptions for this example.

Example Parameters	
UART Data Rate	115,200bps
Address Mode	Extended User
Number of Payload Bytes	100 bytes
Node Processing Time	3ms
AP Processing Time	3ms
Guard Time	3ms
Number of Beacon Bytes	5 bytes

Figure 6: Example Parameters

First calculate t1.

Calculate t1	
Module 1 UART Output	0.434ms
Module 1 Processing Time	3.000ms
t1=	3.434ms

Figure 7: t1 Calculation

The module with the low order byte of 0x01 begins to transmit immediately after receiving the beacon message, and that will take approximately 3.434ms from the start of the first byte. The other modules need to wait for their turn. This wait time is now calculated.

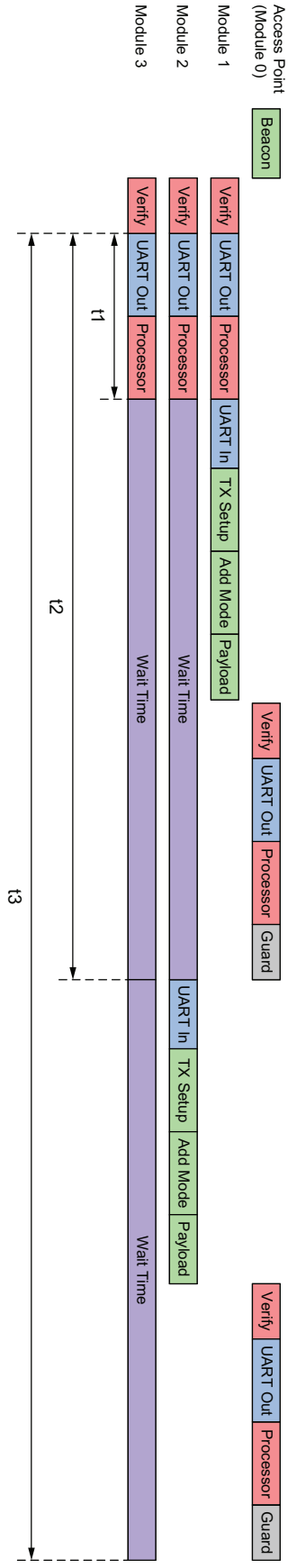
Calculate Wait Time	
Mod UART Input	8.681ms
Mod TX Setup	2.000ms
Mod Address Mode	2.040ms
Mod PPB x # Bytes	6.079ms
Mod0 RX Verify	1.200ms
Mod0 UART Out	8.681ms
Mod0 Processor Time	3.000ms
Guard Time	3.000ms
Wait Time=	34.680ms

Figure 8: Wait Time Calculation

The rest of the start times are calculated using this formula.

t1 + ((Address - 1) * Wait Time)

Figure 5: Module Transmission Timing



Module 1: t1 = 3.434ms
Module 2: t2 = t1 + ((2 - 1) * 34.68 = 38.114ms
Module 3: t3 = t1 + ((3 - 1) * 34.68 = 72.794ms
Module 4: t4 = t1 + ((4 - 1) * 34.68 = 107.474ms

This example can expand to more modules, but the hop timing must be taken into account. If the timing exceeds 400ms at the low RF rate or 390ms at the high RF rate, then the modules need to wait for the next hop. The Access Point should send another beacon immediately after the hop to trigger the next set of modules. If an oscilloscope or logic analyzer is available, the PA_EN line on the module can be used to show the transmission time to verify the timing.

This example starts the timing from the first byte output by the receiving modules. If the responses overlap the hop time, then the beacon message needs to be taken into account. It should be the first message on each hop. As such, it has the extended preamble. This needs to be counted into the beacon transmission time as well as the node receiver verify time.

In this example, the beacon does not need to contain much information. 5 bytes was assumed to contain enough information to trigger the nodes to respond, but it could just as easily be a single byte or more. It depends on what is needed by the system.

Example 2

In this example, the Access Point sends a list of addresses that should respond on the current hop and the time offset for each address. The offset time is calculated as described in Example 1.

This is a better method if there are more modules that need to respond than can fit into a single hop time. This gives the Access Point more control in making sure that it gets data from all modules in the network.

In this case, the beacon message contains the address and offset of each module that should respond.

Example 3

The third example is a basic polling system where the Access Point sends a transmission to each node address requesting a response. The node responds with its data and the Access Point queries the next node in the network. Figure 9 shows an example pseudocode loop for this kind of polling system.

There are two methods to select which node is queried:

1. Change the destination address in the AP to the node's address
2. Send a broadcast message to all nodes and identify the node in the payload.

Both methods work and the selection of the method is driven more by factors external to the module.

Basic Polling Algorithm Pseudocode Loop

```
Do Forever
  Wait for time to poll next node
  Set address for node to be polled
  Send request for data (optional data and poll)
  Wait for response or timeout
  if response received
    Process response
```

Figure 9: Basic Polling Algorithm Pseudocode Loop

In this example, the application does not need to synchronize to the hopping rate, just provide sufficient time for a response that might have to wait for the following slot and use an extended preamble.

This polling method requires many more over-the-air transactions so reduces the throughput, but has the advantage of being more robust. Every node is given direct attention so it is assured that its data comes across, or an error can be thrown to the user.

Other Considerations

If data is to be continuously transmitted from the nodes, then a beacon needs to be transmitted on every hop. This keeps the nodes synchronized to the Access Point so that they can continue to send messages without interruption.

If the payload is normally small, but occasionally needs to be large, the beacon polling can be used to obtain a small status response, and then use a separate simple poll or beacon poll to obtain a larger packet if the earlier status response indicated more data available.

159 Ort Lane, Merlin, OR 97532
Phone: +1 541 471 6256
Fax: +1 541 471 6251
www.linxtechnologies.com