Recommended Procedure for Pulse Sensor CW Measurements

This document presents a basic recommended SCPI command sequence for a Bird pulse sensor to make measurements of CW signals

The Bird pulse sensors are also capable of measurements on complex multiple state pulsed signals. Such measurements require correct configuration of the sensor in order to obtain meaningful results and this is beyond the scope of this document.

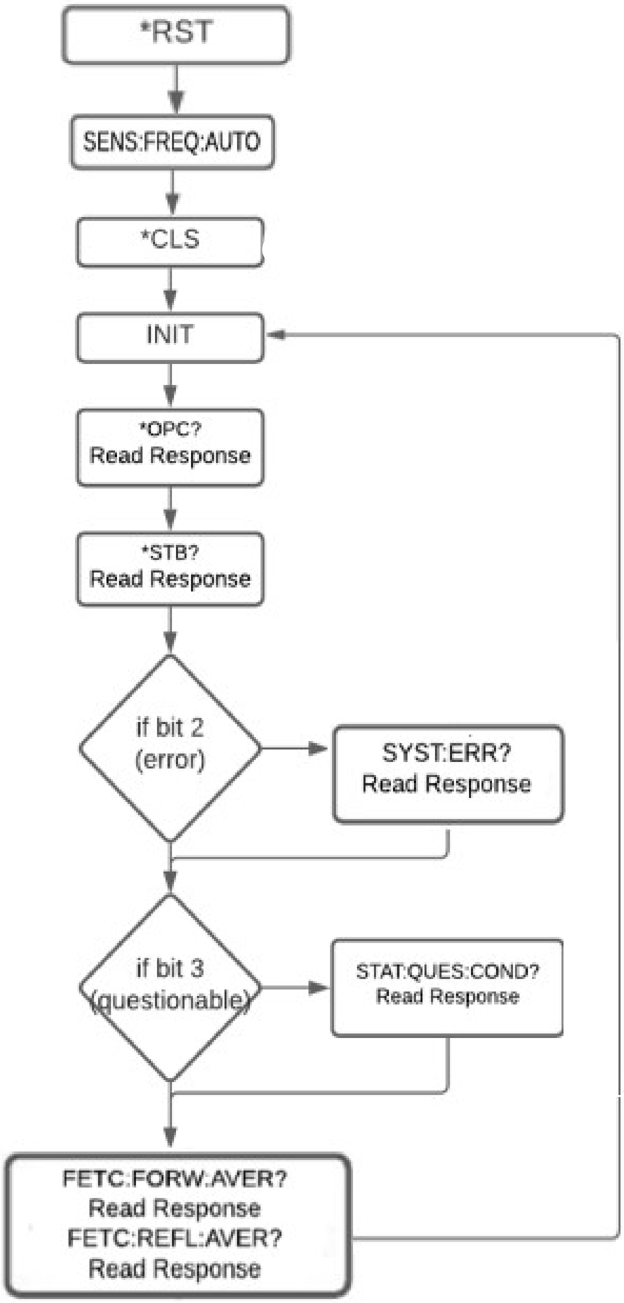
General notes about SCPI

Bird pulse sensors implement a SCPI compliant protocol. The details of SCPI are beyond the scope of this document. For more detail on SCPI itself, see the SCPI standard (https://www.ivifoundation.org/docs/scpi-99.pdf). For a complete reference of the Pulse sensor SPCI commands, see the Programming Manual for your pulse sensor.

Please note the following conventions for SPCI used in this document.

- SCPI commands are divided into Program Messages used to tell the sensor something or Query Messages used to ask the sensor something. Query Messages all end with a question mark ('?').
- Program Messages do not generate a response from the sensor. Query Messages always generate a response. You must issue a read request to read the response to a query message.
- SCPI commands have long and abbreviated forms. This document uses the abbreviated forms. See the Programming Manual for the long versions of the SCPI messages.
- SCPI commands are not case sensitive and can be constructed with arbitrary or even mixed case. In Bird documents, upper case is used to show the minimum abbreviated portions of command words and lower case is used for the remaining long portion of commands.
- SCPI commands are also divided into Common commands and Subsystem commands.
    - Common commands begin with '*' and consist of single word (e.g. *STB?).
    - Subsystem commands are constructed from a list of hierarchical subsystems separated by colons and ending with a program or query item (e.g. SYST:ERR?).
    - Some subsystem items in the hierarchy are optional, and are omitted in this document. Optional items are assumed by default if they are omitted (e.g. INIT in this document is equivalent to TRIG:INIT).
    - See the Programming Manual for your pulse sensor for the full versions of the messages with the optional elements shown.

Figure 1 shows a recommended precure for making a measurement on a CW signal:



Issue the SYST:ERR? query until the SYST:ERR:COUN is 0

You must read the condition register to clear condition bits and allow fresh events on future questionable measurements.

Use the questionable condition bits to determine which measurements are valid to read

Notes on the commands used in this procedure:

- *RST returns the sensor to a known default state.
- SENS:FREQ <freq in MHz> sets a user specified frequency to be used for amplitude calibration in the sensor.
- SENS:FREQ:AUTO <0 or 1> sets the auto frequency mode of the sensor on or off. If auto is on, the sensor will measure the frequency periodically and use this measurement in the amplitude calibration, otherwise the sensor will use the programmed frequency. If the generator is frequency tuning around a known center frequency, programming the frequency will produce more stable measurements and avoid questionable measurements caused by the sensor attempting to measure the frequency while it is moving.
- *CLS clears any prior errors.
- INIT triggers a new measurement (all configured measurements will be performed)
- *OPC? returns a "1" when the prior triggered operation is complete (e.g. a measurement using INIT). Note that the response from the *OPC? query will wait for the operation to complete. This time is dependent on the time needed to complete the pending operation. Long period pulses will result in long measurement times, and the *OPC? query will not complete until the measurement does. For this reason, timeouts on the read operation must allow enough time for the measurement to complete. *OPC? will respond immediately if no prior operation is pending.
- *STB? reads the status byte (see the programming manual for details of the status byte, see table 1 below for the most important bits)
- SYST:ERR? Reads the next system error and reduced the error count.
- STAT:QUES:COND? reads the questionable status condition register and clears any set bits. Note that the questionable bit in the status byte (*STB? Above) aggregates new questionable *events.* Questionable events are triggered by a transition of the bit in the condition register, so if you do not read the condition register to clear the bits, no new events will occur. This behavior is specified in the SCPI standard. See table 2 below for some of the more common questionable conditions. See the Programming Manual for all the details.
- FETC:FORW:AVER? reads the *forward average power* of the RF signal last measured by the sensor. Note that this is not a pulse measurement.
- FETC:REFL:AVER? reads the *reflected average power* of the RF signal last measured by the sensor. Note that this is not a pulse measurement.

Bit	Description	Action
2	Error message queued.	Query the error message(s) with SYST:ERR?
3	Questionable measurement.	Query the questionable condition register with STAT:QUES:COND?
4	Message available.	Read the query response.

Table 1 Status Byte Bits

Bit	Questionable Measurements	Notes
3	Forward Average Power, Reflected Average Power, Gated Average Power	Calibration not valid at measured frequency.
5	Frequency	Insufficient RF amplitude or duration to measure frequency.
8	Forward Average Power, Reflected Average Power, Gated Average Power	Calibration is not valid.
9	Gated Average Power, Pulse Period, Pulse Width, Pulse Repetition Frequency, Pulse Duty Cycle	No pulse detected.
10	Gated Average Power	The begin gate delay and end gate delay exceed the width of the pulse. The begin and end gate overlap.

Table 2 Questionable Condition Register Bits