Rubidium Frequency Standard

model PRS10

Preliminary
Operation and Service Manual

1290 D Reamwood Avenue
Sunnyvale, CA 94089 USA
Phone: (408) 744-9040 • Fax: (408) 744-9049
www.thinkSRS.com •e-mail: info@thinkSRS.com

Revision 1.01 • 1/2000

Stanford Research Systems

Table of Contents

PRS10 Rubidium Frequency Standard

Introduction

The PRS10 is a ultra-low noise 10 MHz frequency standard which disciplines an SC-cut ovenized oscillator to a hyperfine transition in the ground state of rubidium.

The PRS10 was designed to fill a variety of communication, synchronization, and instrumentation requirements. The phase noise of the 10 MHz output is low enough to be used as the reference source for cellular synthesizers. The unit’s short-term stability and low environmental coefficients make it an ideal component in network synchronization systems. Also, the low aging rate makes it an excellent choice as a timebase for precision frequency measurements.

The unit is compatible in fit, form, and function to the Efratom FRS frequency standards, with improvements in features and performance. The PRS10 allows closed case diagnostics and calibration via an RS232 interface, its digital synchronous detection and filtering eliminate spurs on the 10 MHz output, and the PRS10 has 1000x less phase noise than the Efratom unit (-125 dBc vs. –90 dBc at 10 Hz).

The PRS10 can time-tag an external 1 pps input with very high resolution. These values may be reported back via RS232 and/or used to phase lock the unit to an external reference (such as GPS) with a time-constant of several hours. This feature can provide Stratum 1 performance at a very low cost.

In addition to reading time-tag results, the RS232 interface allows the user to set the frequency, adjust the phase of the 1 pps output, read the value of virtually every parameter (lamp drive level, rf level, temperature set point of the crystal, lamp, and resonance cell, and 10 MHz output level) and measure many “test-points” (lamp light level, heater currents, power supply voltages, and case temperature.)

The PRS10 establishes a new level of features and performance in atomic frequency standards. Its design provides for the lowest phase noise and easiest path to system integration of any rubidium frequency standard available.

PRS10 Specifications

		Units
Output		
Frequency	10 (Sine wave into 50Ω)	MHz
Amplitude	0.5 ± 10% (about 1.41V_{pp} or +7 dBm)	V_{rms}
Accuracy	±5x10^{-11} (at shipment)	Δf/f
Allan variance	< 1x10^{-11}(1s), < 1x10^{-11}(10s), < 2x10^{-12}(100s)	Δf/f
SSB phase noise	<-125 (10Hz), <-140 (100Hz), <-150 (1kHz)	dBc/Hz
Spurious	<-130 (100kHz B.W.)	dBc
Harmonics	<-40	dBc
Return loss	> 25 (at 10MHz)	dB
Temperature	±5x10^{-11} over -20°C to +65°C baseplate	Δf/f
Voltage	< 2x10^{-11} for a 1V_{dc} supply change	Δf/f
Magnetic field	< 2x10^{-10} for 1 Gauss field reversal	Δf/f
Retrace	±5x10^{-11} (72 hr off then 72 hr on)	Δf/f
Settability	< 5x10^{-12}	Δf/f
Trim Range	±2x10^{-9}	Δf/f
Time to lock	< 5 (starting at 25°C)	minutes
Time to 1x10^{-9}	< 6 (starting at 25°C)	minutes
Other Electrical		
Power supply	+24.0 (nom), +22 (min), +30 (max)	Vdc
Supply current	2.2 (warmup) , 0.6 (steady-state at 25°C)	A
Protection	± 30 (to any pin except rf output)	V_{dc}
RF protection	100 (stable with any termination)	mA
Ext calibration	0-5.00	V_{dc}
Cal reference out	5.00 ± 0.05	V_{dc}
RS232	9600 (8 bits, no parity, 1 stop bit, 0V/5V levels with x on/x off protocol)	baud
1 pps measurement	±10 (accuracy), ±1 (resolution)	ns
1 pps output set	±10 (accuracy), ±1 (resolution)	ns
Miscellaneous		
Temperature	-20 to +65 (baseplate)	°C
Storage	-55 to +85	°C
Size	2.00 x 3.00 x 4.00	inches
Weight	1.32	lbs
Warranty	2	years
Baseplate threads	4-40 (4 places)	
Connector	Mates with ITT/Cannon DAM11W1S series	

Abridged Command List

Commands consist of two-letter ASCII mnemonics. A command may be followed one or more numeric values, and punctuation. Command sequences end with a carriage return (ASCII 13_{10}). All commands are case insensitive. Spaces (ASCII 32_{10}) and linefeeds (ASCII 10_{10}) are ignored.

A command followed by a value is used to set a parameter to the value. A command followed by an exclamation point (! or ASCII 33_{10}) indicates that the current value should be saved to EEPROM to be used as the initial value after the next reset. A command followed by a question mark (? or ASCII 63_{10}) is used to request that the current value be returned. A command followed by an exclamation point and a question mark is used to return the EEPROM value.

For example, the gain parameter determines the time constant used to lock the 10MHz oscillator to the rubidium hyperfine transition. Examples of the four forms of the gain parameter command are:

```
GA?         ;returns the current value of the frequency lock loop gain parameter.
GA7         ;sets the frequency lock loop gain parameter to 7.
GA!         ;writes the value of the gain parameter to EEPROM for use after reset.
GA!?        ;returns the value of the gain parameter which is stored in EEPROM.
```

All strings returned by the unit are terminated with a carriage return (ASCII 13_{10}). In the verbose mode, strings are preceded with a linefeed (ASCII 10_{10}) and terminated with a carriage return and a linefeed. If more than one value is returned by a command the values will be separated by a comma (ASCII 44_{10}).

When a unit is first turned “on”, it will send the string “PRS_10” (without the quotes) followed by a carriage return.

Only commands in **bold type** are available to the end-user. The other commands are “factory only” commands which disabled at the factory.

Query Value	Set Value or Activate	Write EEPROM	Query EEPROM	Description
Initialize				
RS?	**RS 1**			Restart
VB?	**VB *value***			Verbose mode
ID?				Read ID string
SN?	SN *value*	SN!	**SN!?**	Read unit serial number
ST?				Read six status values
LM?	**LM *value***	**LM!**	**LM!?**	Lock pin mode
	RC 1	RC!		Recall factory calibration
Freq. Lock				
LO?	**LO *value***			Frequency lock loop status
FC?	**FC *high,low***	**FC!**	**FC!?**	Frequency control values
DS?				Read detected signals (ω and 2ω)
SF?	**SF *value***			Set frequency offset
SS?	SS *value*	SS!	**SS!?**	Set Slope (SF calibration)
GA?	**GA *value***	**GA!**	**GA!?**	FLL Gain parameter
PH?	PH *value*	PH!	**PH!?**	Phase angle parameter
SP?	**SP *r,n,a***	**SP!**	**SP!?**	Set synthesizer parameters
Magnetic Tuning				
MS?	**MS *value***			Magnetic switching
MO?	**MO *value***	**MO!**	**MO!?**	Magnetic Offset
MR?				Magnet read
1PPS Lock				
TT?				Time-tag (1 pps input)
TS?	TS *value*	TS!	**TS!?**	Time slope cal. (1 pps input)
TO?	TO *value*	TO!	**TO!?**	Time-tag offset
PP?	**PP *value***			Place pulse (1 pps output)
PS?	PS *value*	PS!	**PS!?**	Pulse slope cal. (1 pps output)
PL?	**PL *value***	**PL!**	**PL!?**	Phase lock (to 1 pps input)
PT?	**PT *value***	**PT!**	**PT!?**	Phase lock time constant
PF?	**PF *value***	**PF!**	**PF!?**	Phase lock stability factor
PI?	**PI *value***			Phase lock integral term

Query Value	Set Value	Write EEPROM	Query EEPROM	Description
D/A Control				
SD0?	SD0,*value*	SD0!	**SD0!?**	Set DAC (RF amplitude)
SD1?	SD1,*value*	SD1!	**SD1!?**	Set DAC (1 pps delay)
SD2?	SD2,*value*	SD2!	**SD2!?**	Set DAC (lamp intensity)
SD3?	SD3,*value*	SD3!	**SD3!?**	Set DAC (lamp temperature)
SD4?	SD4,*value*	SD4!	**SD4!?**	Set DAC (crystal temperature)
SD5?	SD5,*value*	SD5!	**SD5!?**	Set DAC (cell temperature)
SD6?	SD6,*value*	SD6!	**SD6!?**	Set DAC (10 MHz amplitude)
SD7?	SD7,*value*	SD7!	**SD7!?**	Set DAC (RF deviation)
Analog Test	(12bit values)			
AD0?				Spare (J204)
AD1?				+24V(heater supply) / 10.
AD2?				+24V(electronics supply) /10
AD3?				Drain voltage to lamp FET / 10
AD4?				Gate voltage to lamp FET / 10
AD5?				Crystal heater control voltage
AD6?				Resonance cell heater control
AD7?				Discharge lamp heater control
AD8?				Amplified ac photosignal
AD9?				Photocell's I/V converter / 4
AD10?				Case temperature (10 mV/°C)
AD11?				Crystal thermistors
AD12?				Cell thermistors
AD13?				Lamp thermistors
AD14?				Frequency calibration pot
AD15?				Analog ground
Analog Test	(8bit values)			
AD16?				VCXO varactor voltage
AD17?				VCO varactor voltage
AD18?				AGC for RF
AD19?				RF PLL lock signal

Theoretical overview of rubidium frequency standards

Rubidium is an alkali metal (like lithium, sodium, potassium and cesium). There are two naturally occurring isotopes of rubidium, Rb85 and Rb87, which have relative abundances of 72% and 28% respectively. The metal has a melting point of 39°C.

The alkali metals behave similarly: they have one electron outside an inert core. Most of the chemical, electronic and spectroscopic properties of these elements are determined by this outer electron. The deep red glow of a low power rubidium discharge lamp is due to the resonance line transitions of the outer electron as it emits a red photon and drops back to the ground state.

The ground state of Rb87 is split by a very small energy due to the relative orientation of the magnetic spins of the electron and the nucleus. The split corresponds to the energy of a photon with a (microwave) frequency of 6.834,682,612,8 GHz. It is this hyperfine transition frequency which will be used to stabilize the 10 MHz output of the PRS10.

To see how this is might be done, Figure 1 shows a typical physics package which uses a discharge lamp, an isotopic filter, and a resonance cell. We will see that the amount of light which passes through the resonance cell to the photodetector can be reduced when the resonance cell is exposed to microwaves at the hyperfine transition frequency.

To simplify the discussion, we will assume that the light from the Rb87 discharge lamp consists of just two lines corresponding to transitions from a single excited state to the split ground state. The filter cell contains Rb85 vapor which also has a split ground state and an isotopic shift (relative to Rb87) as well. An important coincidence exists: one of the lines from the Rb87 discharge corresponds one of the transitions in Rb85. This will allow us to reduce the intensity of this line by passing the Rb87 discharge light through the Rb85 vapor.

Normally, atoms in the ground state will be equally distributed between the split states, as the splitting is much less than the thermal energy of the atoms in the vapor. This distribution is modified by the filtered light from the discharge, by a process called “optical pumping”.

Suppose that the filter can completely remove one of the two discharge lines. The remaining light can be absorbed by Rb87 atoms in the resonance cell which are in the lower ground state, moving them to the upper state. When they decay from the upper state, they fall with equal probability into either ground state. As this continues, population will be moved from the lower ground state to the upper ground state, circulating through the upper state. As the population in the lower ground state is decreased, the amount of light which reaches the photodetector will increase, as the number of atoms which can absorb the radiation is reduced.

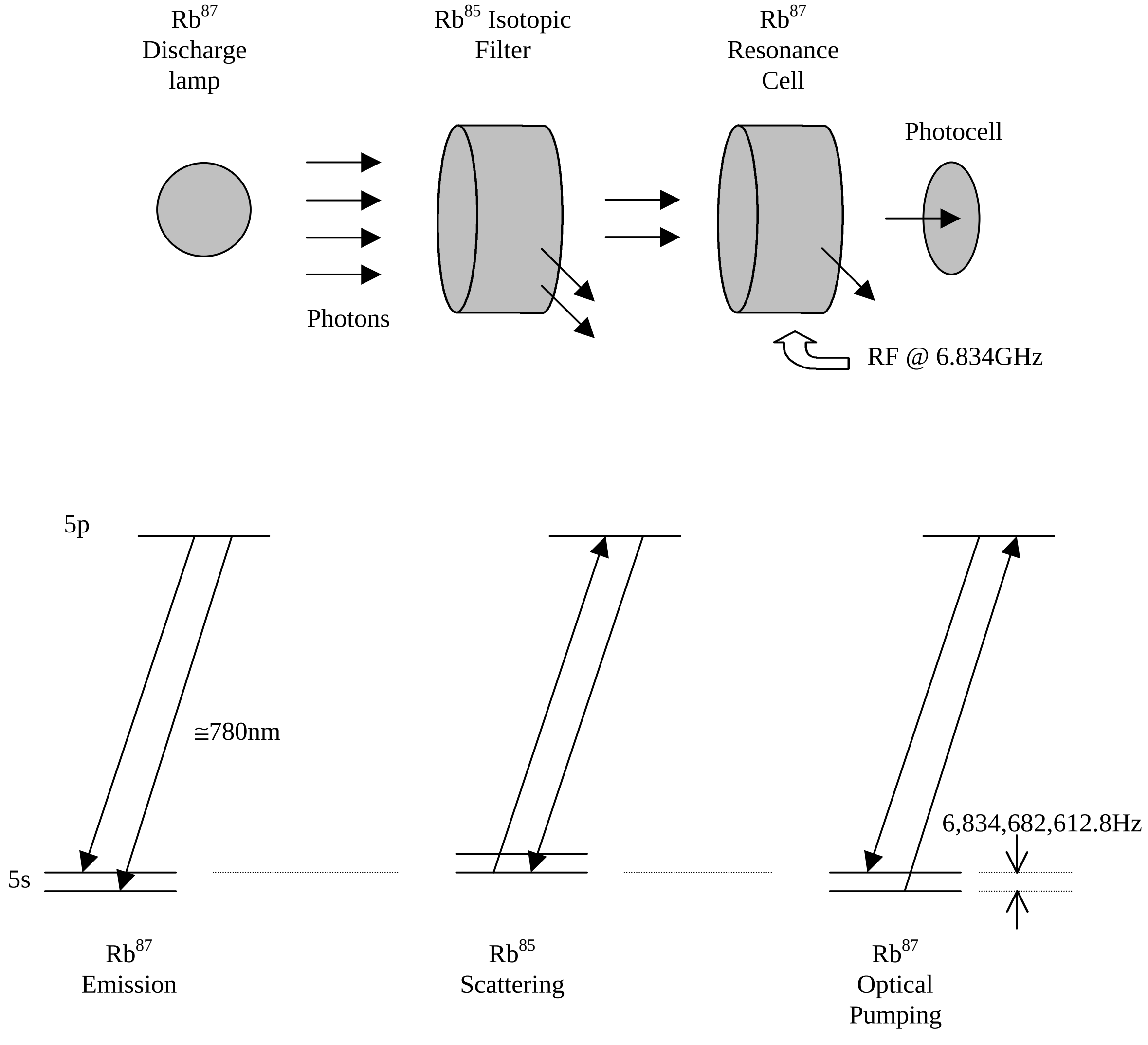

Figure 1. Hypothetical Rubidium Physics Package

Now, if we apply a microwave field at the frequency corresponding to the hyperfine transition frequency (6.834,682,612,8 GHz), the populations in the ground state will mix, and the amount of light reaching the photodetector will decrease.

The PRS10 uses the “integrated filter” topology: rather than a separate filter cell, the resonance cell contains a mixture of the two rubidium isotopes, along with a buffer gas. The lamp also contains a mixture of isotopes. The isotopic mixtures, buffer gases, and operating conditions are chosen so as to minimize temperature coefficients and intensity shifts of the apparent hyperfine transition frequency.

The apparent transition frequency will be shifted by about 3 kHz from the natural transition frequency by the buffer gas and discharge lamp spectral profile. The transition frequency differs slightly for each unit, depending on the fill pressure, etc. The transition frequency is also tuned over a few Hertz by a magnetic field which may be varied.

In the PRS10, the rubidium physics package acts as a very stable frequency detector for a frequency around 6.834 GHz. By using a microwave frequency synthesizer which is referenced to the 10 MHz OCXO, the 10 MHz may be stabilized to the rubidium hyperfine transition frequency.

PRS10 Overview

All compact rubidium frequency standards discipline a crystal oscillator to the hyperfine transition frequency in the ground state of rubidium. Several different topologies have been developed. A major difference in these designs is the method chosen to lock a standard output frequency (usually 10 MHz) to the (essentially arbitrary) hyperfine transition frequency at about 6.834 GHz.

Block Diagram

Figure 2 shows a block diagram for the PRS10 Rubidium Frequency Standard. The design of the PRS10 is quite different from other rubidium frequency standards leading to several feature and performance benefits.

Ovenized Oscillator

The output from PRS10 comes directly from a 10 MHz oven stabilized, 3rd overtone, varactor tuned, SC-cut crystal oscillator. The varactor is tuned by a 22bit digital-to-analog converter which provides a full scale tuning range of ±2 ppm. The very fine step size ($\approx 1:10^{-12}$) maintains the low noise inherent to the SC-cut resonator, yet the full-scale range is sufficient to compensate for crystal aging over the lifetime of the unit. This approach provides a 10 MHz output with extremely low phase noise which is virtually free of spurs.

Frequency Synthesizer

The 10 MHz also serves as the reference source to the frequency synthesizer which generates RF at about 359.72 MHz. The RF is multiplied by a factor of 19x in a step recovery diode to provide the microwave frequency (at about 6.834 GHz) which is used to interrogate the physics package. (The apparent hyperfine transition frequency varies with each physic package due to variations in buffer gas fill pressure, etc.)

The frequency synthesizer has two important characteristics: a step size of about $1:10^{-9}$ and very low phase noise output. The small step size is required so that only small magnetic fields will be needed to tune the apparent hyperfine transition frequency between the steps of the synthesizer. The low phase noise is required so as not to degrade the signal from the physics package, which would lead to a noisy frequency lock, and degraded output stability.

These two characteristics require a dual loop design for the frequency synthesizer. The inner loop consists of the 359.72 MHz VCO which is directly phase locked to a 3rd overtone 22.48252 MHz crystal oscillator. This loop has a large natural frequency of about 400,000 r/s. The VCO’s phase noise at 359.72 MHz is very close to the phase noise of the crystal (plus 24 dB for the multiplication factor of 16).

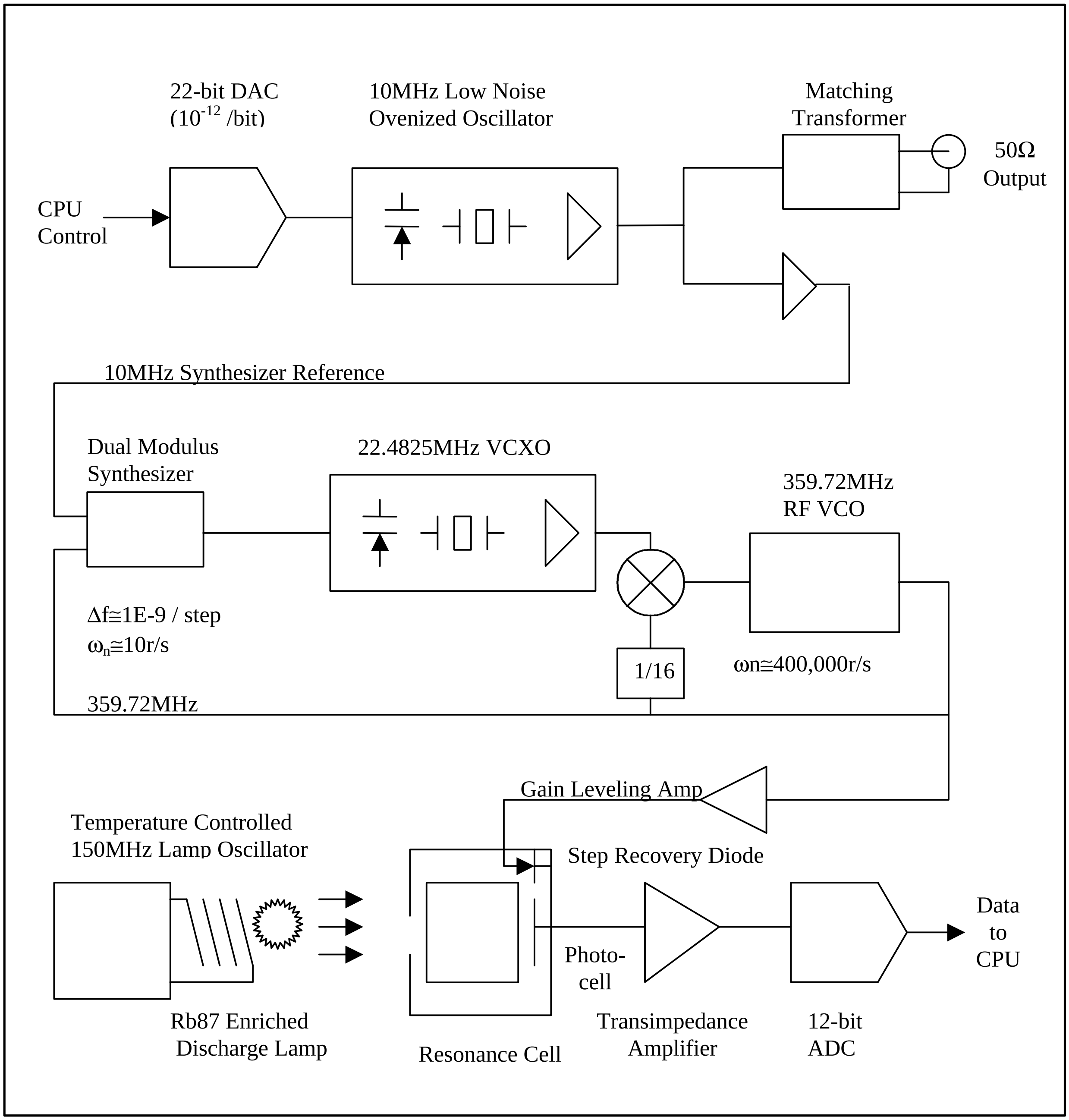

Figure 2. Rubidium Frequency Standard Block Diagram

The outer loop compares the RF frequency to the 10 MHz. This loop provides high resolution by dividing the RF and 10 MHz by large numbers, and consequently operates at a low comparison rate (typically 4 kHz). This loop has a low natural frequency (about 10 r/s) so the phase noise of the RF more than a few Hz from carrier will be determined by the inner loop. The outer loop slowly disciplines the frequency of the inner loop's crystal, keeping it locked to the 10 MHz reference.

The frequency synthesizer is set to the nearest frequency above the apparent hyperfine transition for the unit's physics package. A magnetic field is used to tune the physics package's apparent hyperfine transition frequency up to the synthesizer frequency. A 70 Hz digitally synthesized sine wave is used to phase modulate the inner loop. (The outer loop bandwidth is too small to suppress this modulation.) This generates an RF output, which when multiplied to 6.834 GHz, sweeps by about 300 Hz around the apparent hyperfine transition frequency. By sweeping through the transition at 70 Hz, the output from the photocell will have an ac component at 140 Hz, when centered on the transition. There will be an ac component at 70 Hz if we are off to one side of the transition: the phase of the 70 Hz component is used to determine if the RF is above or below the transition.

Physics Package

The physics package consists of a discharge lamp (enriched with Rb87) and an integrated filter and resonance cell. The discharge lamp operates at about 150 MHz. The lamp oscillator can provide up to 300 Vpp to start the lamp, which drops to about 100 Vpp during normal operation. The lamp oscillator voltage and current are carefully regulated to provide a consistent intensity and low noise.

The resonance cell is inside a mu-metal shell to reduce the frequency pulling effects of external magnetic fields. The apparent hyperfine transition frequency may be quadratically tuned over a range of about $\pm 2 \times 10^{-9}$ by the magnetic field coil. (The frequency shift is always positive, regardless of the direction of the magnetic field.)

To further reduce the effects of external magnetic fields, the current in the field coil is switched at 5 Hz. An external field which adds to the coil's field will increase the apparent transition frequency, and an external field which opposes the coil's field will decrease it. By alternating the coil's field and averaging, the effect of an external field can be reduced.

Control Algorithm

The microcontroller is responsible for (1) generating the 70 Hz phase modulation of the RF to probe the physics package, (2) synchronously detecting the amplitude and phase of the photosignals at 70 Hz and 140 Hz, and (3) digitally filtering the error signal to lock the 10 MHz SC-cut ovenized oscillator to the rubidium hyperfine transition.

The 70 Hz digitally synthesized phase modulation waveform is generated via a 12bit DAC in 32 discrete steps. A low pass filter is used to remove image frequencies from the modulation waveform. The microcontroller’s hardware timers are used synchronize updating of the DAC so as to eliminate sample jitter. The modulation waveform has very little distortion, noise or spurs, and is precisely 70 Hz.

The photosignal is amplified and bandpass filtered before being converted by a 12bit ADC. The microcontroller multiplies the ADC samples by table data corresponding to sines and cosines at 70 Hz and 140 Hz. The products are summed over a frame of 14 modulation cycles which completely eliminates signal components at 5 Hz, (and at any integer multiple of 5 Hz including 50 Hz, 60 Hz, 70 Hz and 140 Hz) from the error signal, so that there will be no spurs at the modulation frequency in the 10 MHz output.

The summed product corresponding to the detected signal at 70 Hz and 0° is used to frequency lock the 10 MHz oscillator to the Rb hyperfine transition frequency. This value is filtered in a simple, first order, IIR digital filter. The filter coefficient determines the frequency lock loop time constant. Time constants from 1 s to 128 s are available to optimize the output stability of the 10 MHz.

Initial Locking

When power is first applied to the unit, the EFC (the electronic frequency control, or, the voltage applied to the varactor in the 10 MHz SC-cut oscillator) is set to the last value for which the unit was locked. As the 10 MHz oscillator heats to its operating temperature, the output frequency will increase smoothly to converge on 10 MHz. In most cases, the output frequency will be within 0.1 Hz of 10 MHz even before the lock to rubidium is achieved.

After the lamp starts, and the physics package settles to its operating temperature, a resonance signal will be detected by the processor, and used to lock the crystal oscillator to rubidium. In the case that no signal is detected, or if the signal is lost during normal operation, the processor will suspend the frequency lock loop, and maintain the varactor voltage to the 10 MHz ovenized oscillator at a fixed level. Any of the following conditions would cause the CPU to suspend lock:

1) The detected signal at 140Hz is very low.
2) The discharge lamp light level is outside an acceptable range.
3) The RF synthesizer is unlocked.

4) The RF AGC level is pinned high or low.
5) The VXCO varactor voltage is outside the acceptable range.

Suspending lock will prevent a radical change in output frequency in the case of a physics package failure. So, in the case of most failures which cause loss of the lock to rubidium, the 10 MHz will maintain a stable output, with an aging of a few parts in 10^{10} per day.

Locking to External 1pps

The PRS10 may be locked to an external 1 pps source (from a GPS or LORAN receiver, for example) by applying a 1 pps pulse to the 1 pps input (pin 5 on the main connector). A second order digital phase lock loop (PLL) is used to adjust the frequency of the PRS10 to match the frequency of the 1 pps source over long time intervals.

The block diagram of this PLL is shown in Figure 3. The "phase detector" is the time-tagging circuit and firmware, which has a gain of K_{det} = 1bit/ns. The loop filter is a digital filter consisting of an optional prefilter and a standard proportional-integral controller (PI controller) with programmable proportional and integral gains. The VCO is the rubidium frequency standard, whose frequency, f, is tuned by the magnetic field via the SF command parameter with a sensitivity for its 1 pps output of K_{vco} = 0.001 ns / bit-s, or (1 part in 10^{12}) / bit. The response function for each of the elements of the digital PLL is also indicated in the figure in terms of the standard Laplace variable *s*.

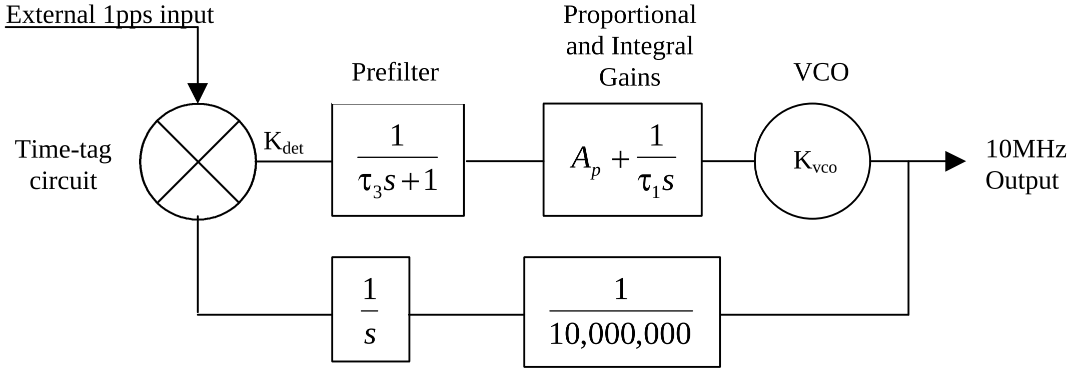

Figure 3. External 1pps Phase Locking Block Diagram

The PI controller is programmed by choosing an appropriate integrator time constant, τ_1, and a stability factor, ζ, which, together with τ_1, determines the proportional gain, A_p, of the controller. τ_1 determines the natural time constant, τ_n, of the PLL for following a step in phase of the reference, while ζ determines the relative rise time and ringing of the PLL in response to the step. The value of ζ also represents the tradeoff in the equivalent noise bandwidth verses peaking in the passband near the natural frequency of the response function.

The PRS10 accepts integrator time constants, τ_1, ranging from 2^8 to 2^{22} seconds in powers of 2. The natural time constant is given by $\tau_n = \sqrt{\tau_1 / K_{\text{det}} K_{vco}} = \sqrt{(1000\text{s})\tau_1}$. Thus, the PRS10 provides natural time constants ranging from 506 seconds to 18.0 hours.

The PRS10 accepts stability factors ranging from 0.25 to 4.0 in powers of 2. The default value of $\zeta = 1.0$ corresponds to a critically damped response; $\zeta < 1.0$ and $\zeta > 1.0$ correspond to underdamped and overdamped responses respectively.

With τ_1 and ζ specified, the proportional gain, A_p, of the controller is given by the equation $A_p = 2\zeta / \sqrt{K_{\text{det}} K_{vco} \tau_1} = 2\zeta / \sqrt{(0.001\text{s}^{-1})\tau_1}$. With the default time constant, τ_1, of 65,536 seconds and a stability factor, ζ, of 1.0, the proportional gain will be about 0.25. In this case the instantaneous frequency of the rubidium source will be adjusted by about 0.25 parts in 10^{12} per nanosecond of time-tag measured.

The PRS10 also provides an optional prefilter. The prefilter is enabled by default, but it can be disabled by sending the command **LM0**, which puts the PRS10 into lock mode 0. When the prefilter is enabled, the PRS10 will exponentially average the time tags output by the "phase detector" before passing the result to the PI controller. The time constant of the prefilter, τ_3, is hard coded to be $\tau_n/6.0$ in order to obtain the maximum benefits of the averaging while simultaneously insuring that the PLL will be stable

Use of the prefilter is recommended when locking to references that have poorer short term stability than the PRS10, but better long term stability. Locking to the 1 pps output by GPS is a prime example of such a case. Use of the prefilter dramatically reduces the digital PLL's sensitivity to the sort term jitter of 50 to 300 ns present on the GPS reference 1pps. The GPS reference also has a significant amount of 1/f noise associated with it. Very long time constants are therefore required to prevent the PRS10 from following this noise too closely. The PRS10 provides natural time constants of up to 18.0 hours, which will allow the PRS10 to follow GPS over time scales on the order of a day, but retain the superior short term stability of the rubidium clock. When locking to a reference that has short term stability comparable to the PRS10, disabling the prefilter is recommended because it will allow the PRS10 to better track the phase of the reference.

In lock mode 0, the PRS10's digital PLL will approximate one of the following three equations depending on the value of ζ:

$$\Delta T(t) = \frac{F_0 - \zeta \Delta T(0)/\tau_n}{\sqrt{1-\zeta^2}/\tau_n} e^{\frac{-\zeta t}{\tau_n}} \sin(\sqrt{1-\zeta^2}\, t/\tau_n) + \Delta T(0) e^{\frac{-\zeta t}{\tau_n}} \cos(\sqrt{1-\zeta^2}\, t/\tau_n) \qquad \text{for } \zeta < 1$$

$$\Delta T(t) = t[F_0 - \Delta T(0)/\tau_n] e^{\frac{-t}{\tau_n}} + \Delta T(0) e^{\frac{-t}{\tau_n}} \qquad \text{for } \zeta = 1$$

$$\Delta T(t) = \frac{-\left[F_0 - (\zeta + \sqrt{\zeta^2 - 1})\Delta T(0)/\tau_n\right]}{2\sqrt{\zeta^2 - 1}/\tau_n} e^{\frac{-(\zeta + \sqrt{\zeta^2 - 1})t}{\tau_n}} + \frac{\left[F_0 - (\zeta - \sqrt{\zeta^2 - 1})\Delta T(0)/\tau_n\right]}{2\sqrt{\zeta^2 - 1}/\tau_n} e^{\frac{-(\zeta - \sqrt{\zeta^2 - 1})t}{\tau_n}} \qquad \text{for } \zeta > 1$$

$\Delta T(0)$ is the initial offset in phase of the PRS10 from the reference. F_0 is the initial offset in frequency of the PRS10 from the reference. $\Delta T(t)$ details how the PRS10 approaches the phase of the reference as a function of time. With the default time constant, τ_1 = 65,536s, and stability factor, ζ = 1, the PRS10's 1 pps output will exponentially approach the phase of the reference 1 pps input with a time constant τ_n = 8,095 seconds or approximately 2¼ hours. In lock mode 1, the equations describing $\Delta T(t)$ are qualitatively similar to those presented above, but generally can only be solved numerically.

The locking algorithm of the PRS10 proceeds as follows:

- The 1 pps PLL is enabled when the unit is turned-on or restarted if the PL parameter stored in the unit's EEPROM is "1".

- The PLL will begin to control the frequency of the rubidium frequency standard when 256 consecutive "good" 1 pps inputs (i.e., 1 pps inputs which are within ±2048 ns of the first time-tag result, modulo 1 s) are received.

- After receiving 256 consecutive "good" 1 pps inputs, the 1pps pulse delay is set to the last of the 256 time-tag values. (For example, if the last of the 256 "good" time tag values is 123,456,789 ns then the program will set the 1 pps output delay to 123,456,789 ns, which moves the 1 pps output by 123,456,789 ns, so that new time-tag values will be about zero.) Also, the current value of the SF parameter (which adjusts the frequency of the rubidium frequency standard over the range of ± 2000 parts in 10^{12}) is used to initialize the integrator, Int(0). (The current value of the SF parameter may be from the internal calibration pot position, an external calibration voltage, the value from a previously received SF command, or the value left over from a previous PLL lock.) If the prefilter is enabled, the exponential filter for the time tags is zeroed.

• The unit will lock the frequency of the PRS10 to the “good” 1 pps input pulses. “Bad” 1 pps inputs (1 pps inputs with time-tags greater than 1,024 ns from the last “good” 1 pps input) will be rejected. The frequency parameter, f, to the SF command will be updated with each “good” time-tag result, $\Delta T(n)$, as follows:

The prefilter : if **LM0** $\overline{\Delta T}(n+1) = \Delta T(n)$

The prefilter : if **LM1** $\overline{\Delta T}(n+1) = (1.0 - \Delta t/\tau_3)\overline{\Delta T}(n) + (\Delta t / \tau_3) \Delta T(n)$

The integral term: $Int(n+1) = Int(n) - (\overline{\Delta T}(n+1) / \tau_1)K_{det}\Delta t$

The proportional term: $Pro(n+1) = -A_p \overline{\Delta T}(n+1)K_{det}$

The frequency setting: $f(n+1) = Pro(n+1) + Int(n+1)$

In the above equations, Δt is the time between phase comparisons, which is one second for the PRS10. The frequency control value, f, ranges over ±2000 bits. If the new f value exceeds 2000, it is set to 2000. If the new f value is less than -2000, it is set to - 2000.

If the new integral term exceeds 2000, it is set to 2000. If the new integral term is less than -2000, it is set to -2000. This will prevent “integrator wind-up” in the case that the f-value is pinned for a long time to slew the 1 pps output in line with the 1 pps input.

The output of the digital filter, f, is used as the frequency control parameter for the SF (set frequency) command, which is updated once a second.

• The PLL will be aborted and restarted if there are 256 consecutive “bad” 1 pps inputs. (This could happen if the 1 pps input is moved suddenly by more than 1,024 ns.) The PLL will also be aborted and restarted if the measured time-tag value for a “good” 1 pps input exceeds ±4 ns/s * τ_1. (For τ_1’s default value of 65,536 seconds, the PLL will restart if the absolute value of a “good” time-tag is greater than 262,144 ns. This could happen if the 1 pps input is more than a few parts in 10^{-9} off the correct frequency for a long time.)

CPU Tasks

In addition to the frequency lock loop control, the microprocessor is responsible for a variety of other tasks. The CPU sets D/A values which control the microwave amplitude, the lamp intensity, the 10 MHz output amplitude, and set the temperature of the crystal, lamp and resonance cell. The CPU will also controls peripheral electronics to output a 1 pps pulse (with 1ns placement) and measure the time for a 1 pps input pulse (with 1 ns resolution).

There is an RS232 interface which allows closed-case calibration of the PRS10. This capability may also be used to servo the 10 MHz or 1 pps outputs to another frequency or time source in a system. For example, this would allow the PRS10 to be locked to the 1 pps from a GPS receiver with a long time constant to eliminate aging.

PRS10 Applications

In virtually all cases, the PRS10 may be “dropped into” applications which use the Efratom FRS-C-1A8A4C (10 MHz sine output, -5°C to +65°C) or the FRS-N-1A8A4B (10 MHz sine output, -55°C to +65°C).

Some customers may wish to evaluate the PRS10 on the bench. To facilitate this, SRS can provide a connector adapter, power supply and RS232 cable. The adapter breaks-out the Cannon plug on the PRS10 to a power connector (2.1 mm with +24 V to center pin), three BNCs (10 MHz and 1 pps output and 1 pps input), and a DB9 (for the RS232). The adapter also has status indicators for power, lock and RS232 activity. This kit allows the PRS10 to be operated from 110-240 Vac (50/60 Hz), provides for a direct connection to a PC via a serial port (typically COM2:), and allows the use of standard BNC cables.

The PRS10 may also be operated with a customer supplied connector (Cannon series DAM11W1S with coaxial insert DM53740-5008 for RG174 cable) from a bench dc power supply. The power supply should be able to supply 2.2 A at +24 Vdc.

Interface Connector

Pin	Name	Description
1	LOCK/1PPS	Lock indication and 1pps output. (See LM command)
2	POT WIPER	Ext. freq. calibration. Nom:+2.50 V. 0-5 V for ±2x10^{-9}
3	POT -	Ground reference for external frequency calibration.
4	TXD/PHOTO	RS232 data output or photo I/V monitor output
5	1PPS_IN/PHOTO	1pps input for time-tagging or photo-amp output
6	+24(HEAT)	+24 Vdc supply for discharge lamp and heaters
7	RXD/EFC	RS232 data input or EFC monitor output
8	POT +	+5.00 Vdc reference output for external freq. cal. pot.
9	+24(CLEAN)	+24 Vdc supply for electronics (not heaters or lamp)
10	GROUND	Case ground and power supply return
coax	10MHz	10 MHz sine output on center conductor

Configuration Notes

The functions of three pins (4,5 and 7) on the interface connector may be modified by internal hardware jumpers. The function of the LOCK output may be modified via RS232.

Pin 1: LOCK/1PPS output. The default configuration is: +5 V indicates that the unit is not locked to rubidium (as during warm-up), 0 V indicates a successful lock of the 10 MHz oscillator to rubidium, pulsing high for 10 μs at a 1 pps rate. The 1 pps output may be moved earlier by any interval from 1ns to 999,999,999 ns via RS232 command. The unit may be configured to omit the 1 pps output via the LM command (via RS232).

Pin 4: TXD/PHOTO The default configuration uses this pin as an output for RS232 data. Many system parameters (including the lamp intensity) may be monitored via the RS232 interface. The function of this pin may be changed to an analog monitor for the lamp intensity by removing one resistor (R347) and installing a 10 kΩ resistor for another (R348) on the microcontroller PCB.

Pin 7: RXD/EFC The default configuration uses this pin as an input for RS232 data. Many system parameters (including the EFC, electronic frequency control) may be monitored via the RS232 interface. The function of this pin may be changed to an analog monitor for the EFC by removing one resistor (R354) and installing a 10 kΩ resistor for another (R353) on the microcontroller PCB.

Pin 5: 1PPS_IN/PHOTO The default configuration uses this pin as a 1 pps input to allow time-tagging or phase locking to an external 1 pps source. The function of this pin may be changed to allow monitoring of the amplified photo-signal. When configured as a 1PPS_IN, R241 will be omitted on the top PCB, and a 1 kΩ resistor will be installed for R242. When configured for PHOTO_AMP output, R242 will be omitted on the top PCB, and a 1 kΩ resistor will be installed for R241.

10 MHz coax shield The default configuration floats the shield of the 10 MHz coaxial connector with respect to ground. The 10 MHz output is transformer coupled, and the shield may be ground referenced by installing the jumper between J101 and J102 (located near the connector on the 10 MHz oscillator PCB.)

Hardware Notes

All of the pins on the interface connector are protected against continuous connection to any potential up to 24 Vdc. The power supply pins are protected against polarity reversal and may be operated up to +30 Vdc. In most applications, both +24 Vdc supplies (heater and electronic supplies) will be connected together and operated from a +24 Vdc supply.

Logic outputs (LOCK/1PPS and TXD/PHOTO) have a 1 kΩ output resistance driven by a CMOS logic device operating between +5 Vdc and ground. Logic inputs (RXD/EFC and 1PPS_IN) have 100 kΩ to ground and 3.9 kΩ CMOS gate inputs (which have input protection diodes to +5 V and ground).

RS232 data is sent to the host on pin 4, received from the host on pin 7. The baud rate is fixed at 9600 baud, 8 bits, no parity, with 1 start and 2 stop bits. No DTR or CTS controls have been used, rather, the XON/XOFF protocol has been implemented. The transmit drive level is 0 and 5 V, not the +/-12 V normally associated with RS232. These levels are compatible with RS232 line receivers, but does not require their use (a

TTL inverter may be used instead), hence simplifies the interface when used inside an instrument at the sacrifice of degraded noise immunity over long lines.

The PRS10 may be connected directly to a PC’s COM2: port with three wires: TXD, RXD, and ground. As the PRS10 sources only +5/0 V for the RS232 (via 1 kΩ) the connecting cable should be kept short.

PRS10	PC’s COM: (DB9 Connector)	PC’s COM: (DB25 Connector)
Pin 7 (RXD)	Pin 3 (TXD)	Pin 3 (TXD)
Pin 4 (TXD)	Pin 2 (RXD)	Pin 2 (RXD)
Pin 10 (GND)	Pin 5 (GND)	Pin 7 (GND)

Operating Temperature

The unit should be operated so that the baseplate temperature stays below +65 °C. This requirement is usually met by units operating on the bench at room temperature when powered by +24 Vdc.

Frequency Adjustment

A magnetic field coil inside the resonance cell is used to tune the hyperfine transition frequency. The magnetic field is controlled by a 12bit DAC. The output frequency (at 10 MHz) tunes quadratically with the DAC setting ($0 \leq \text{DAC} \leq 4095$), and $\Delta f(\text{Hz}) \approx 5\text{x}10^{-9} \text{ x DAC}^2$. The DAC setting is changed from the nominal calibration value (see MO command) in various ways, including, calibration pot position, external calibration voltage, direct setting (see SF command), and external 1 pps PLL control.

When the unit is first turned on (or restarted), the internal frequency calibration pot position will be used to set the DAC relative to the calibration value stored in EEPROM. (If a voltage is applied to pin 2 of J100 (POT_W) then this voltage will override the pot position.) An SF command may be sent, or a 1 pps input may be applied, to control the frequency offset directly. If either the SF command or the 1pps input control the frequency offset, then the pot position (or external control voltage) will not be used again until the power is cycled or the unit is restarted. All the various ways to adjust the frequency of the 10 MHz output are linearized, and they have a span of $\pm 2000 \text{ x } 10^{-12}$ or ±0.020 Hz.

RS232 Instruction Set
Syntax

Commands consist of a two letter mnemonic and one or more parameters. Commands which end with a question mark (?) will return a value. Commands which end with an exclamation point (!) write the current parameter value to EEPROM for use after the next restart. Commands which end in an exclamation point and a question mark (!?) return the value stored in EEPROM.

All data is communicated in ASCII codes. Commands are case insensitive, and spaces (ASCII 32_{10}) are ignored. Commands are processed when a carriage return (ASCII13_{10}) is received. Returned values are delimited with commas (in the case of multiple returned values) or a carriage return (in the case of a single or the last returned value). Command available to the end user are in **bold**: some commands are for factory use only and a special code must be transmitted to enable these commands. Parameter lists are enclosed in curly brackets { }, the brackets are not part of the command.

On reset, the unit will transmit the characters: PRS_10 with a carriage return.

Initialization

RS 1

Restart. This command will restart the PRS10's microcontroller just like power-on. (It is not necessary to send a **RS** command on power-up.) All values will return to the values stored in EEPROM. (verbose mode disabled, 10 MHz set to last stored value, etc.) The frequency lock-loop will be disabled until the microcontroller verifies that the unit is warmed-up and that a useful signal level is present. Example**: RS 1** will cause the unit to restart.

VB{0 or 1}
VB?

Set verbose mode. The verbose mode is useful when a human is communicating with the PRS10 using a terminal program (the PRS10 will provide an "OK", command prompts, etc.) The verbose mode should be disabled when a computer program is communicating with the PRS10 (where format characters would interfere.) **Examples: VB0** disables the verbose mode (this is the power-on default mode.) **VB1** will enable the verbose mode.

ID?

Identify. This command returns an identification string which includes the serial number and firmware version of the PRS10. **Example: ID?** will return the identification string PRS10_3.15_SN_12345 (model _firmware version_serial-number).

SN?
SN{value}
SN!
SN!?

Serial number. This command returns the unit's serial number. **Example: SN?** Will return 21567 for a unit with serial number 21567. The command to write and burn a serial number are for factory use only.

ST?

Status. This command will return a six number string corresponding to the values of the six status bytes. Each number will range between 0 and 255, and will be separated by commas. (For definitions of the status bytes, refer to the end of the detailed command descriptions.)

LM{value} value = 0, 1, 2, or 3
LM?
LM!
LM!?

Lock mode pin configuration. This command is used to configure the LOCK/1PPS output (pin 1 on the main connector, J100.) The LOCK/1PPS pin may be configured per the following table:

LM	Description of LOCK/1pps Output
0	Output goes low when locked to Rb, pulses high for 10 µs at 1 Hz 1 pps locking prefilter disabled
1	Output goes low when locked to Rb, pulses high for 10 µs at 1 Hz 1 pps locking prefilter enabled (default)
2	Output goes low when locked to Rb, 1pps is disabled
3	Output goes high when locked to Rb, 1pps is disabled

The default value is 1, so that pin 1 will go low when the unit is locked to rubidium, and will pulse high for 10 µs at a 1 Hz rate. The position of the 1 pps pulse may be moved with the PP command. **Example: LM?** Could return 1, indicating that the unit is in its default configuration so that the lock pin goes low when locked to Rb, pulsing high for 10 µs at a 1 Hz rate. To configure the unit for no 1 pps output, the command string **LM 2** followed by **LM !** will change the unit's power on default for no 1 pps output.

RC 1
RC!

Recall. This command is used to return all values in EEPROM to the values which were present when the unit was first shipped from the factory (except for the unit-start and lamp-start counters. This command should be used if you have been writing values to EEPROM and have somehow corrupted the operation of the device. Executing this command may require calibration of the unit, as the frequency calibration values are also returned to their factory values. The unit will be restarted after the values in EEPROM have been restored to their factory values. **Example: RC 1** will return all calibration values to the values which were determined for the unit when it was manufactured and restart the unit. The RC! command is a factory-only command which writes all of the current parameter values to the EEPROM.

Frequency Lock-loop Parameters

LO{value} value = 0 or 1
LO?

Lock. This command can be used to stop the frequency lock-loop (FLL). It is essentially the same as setting the gain parameter to zero. It may be desirable in a particular application to stop the FLL and set the frequency control value for the 10MHz oscillator manually. (See the FC command.) **Example: LO 0** will stop the FLL. **LO?** will return a value of 0 (if the FLL is not active) or 1 (if the FLL is active.)

FC?
FC{high,low} 0 ≤ high ≤ 4095 1024 ≤ low ≤ 3072
FC!
FC!?

Frequency control. These commands allow direct control of the 22bit value which controls the frequency of the 10 MHz ovenized oscillator. Normally, this value is controlled by the FLL control algorithm, however, the FLL may be stopped, and the value adjusted manually. (See the LO command.)

Two 12bit DACs are scaled (by 1000:1) and summed to provide a varactor voltage which controls the frequency of the 10 MHz oscillator. The low DAC, which operates over half its range (to avoid FFL oscillations at the roll-over to the high DAC) provides a LSB frequency resolution of 1.5:10^{-12}. The high DAC, which has a nominal value of

2048, has a LBS resolution of $1.5{:}10^{-9}$. These DACs provide a total tuning range of about ±3 ppm.

Example: Suppose a unit's FLL has been operating for some time and has settled. An **FC?** will return the current value of the DAC pair which might be 2021,1654. (Tracking the FC value over a long period of time tells you about the frequency variations of the 10MHz crystal. The FC values will change to correct for variations in the crystal frequency due to aging and ambient conditions.)

Both DACs may be set to any value in the range specified above. **Example: FC 2048,2048** will set the 10MHz oscillator back to the middle of its tuning range. However, it is possible to set the frequency of the 10 MHz oscillator so far from the correct frequency that the FLL signal disappears, making the lock impossible. If this happens, the last saved FC value may be read from EEPROM with the **FC!?** command and restored with the **FC{high,low}** command.

The **FC!** command is used to save the current FC values in the unit's EEPROM. The **FC!?** Command may be used to read the value which is stored in the EEPROM. The value stored in EEPROM is used to set the 10 MHz at startup, before the FLL can be established. Occasionally while the unit is operating (at about 20 minutes after power-on and once a day there after) the program will write a new value to EEPROM to correct the value for crystal aging. **Example: FC!?** will return four values (separated by commas), the number of power cycles the unit has undergone, the number of times the FC pair has been written to EEPROM, and the value of the FC pair (high, low) which is used at turn-on and restart.

DS?

Detected signals. This command returns two numbers corresponding to the synchronously detected signals at the modulation frequency, ω_{mod}, and at twice the modulation frequency, $2\omega_{mod}$.

The first number, the amplitude of the signal at ω_{mod}, is the error signal in the rubidium frequency lock loop. The value is proportional to the instantaneous frequency error of the 10 MHz oscillator as detected by the physics package. The value may be large when the unit is first locking, and will bobble around zero in steady state. Each LSB corresponds to about 15 µVrms of signal at ω_{mod}.

The second number is the amplitude (in millivolts rms) of the synchronously detected signal at twice the modulation frequency, $2\omega_{mod}$. The amplitude of this signal is proportional to the strength of the rubidium hyperfine transition signal.

The returned value is a spot measurement taken over just one cycle of the modulation frequency. Since the signals have several Hz of equivalent noise bandwidth, they will be rather noisy.

Example: DS? could return 55,800 indicating a small error signal and a strong resonance signal.

SF{value} -2000 ≤ value ≤ +2000
SF?

Set frequency. This command is used to override the internal calibration pot (or external calibration voltage) to set the frequency directly, relative to the calibration values in EEPROM (see the SP and MO commands.) The command sets the frequency offset in units of parts in 10^{-12} (corresponding to a frequency resolution of 10 μHz at 10 MHz.) The SF? command will return the currently set relative frequency value (with a range of ±2000) whether the value comes from the internal calibration pot position, an external frequency control voltage, an SF command, or from the external 1 pps phase lock loop control algorithm. However, SF set command is ignored if the unit is phase-locked to an external 1 pps signal. (To re-establish direct control via the SF command, the PLL must be disabled. See PL 0 command.)

Example: SF 100 will set the frequency 100 x 10^{-12} (or 0.001 Hz) above the stored calibration value, and the **SF?** command will return 100.

Data from the SF command cannot be saved when the power is turned off. (To do this type of calibration, see the SP and MO commands.) Once executed, the SF command will disable the analog channels (internal calibration pot and external calibration voltage) until the power is cycled or the unit is restarted.

SS?
SS{value} 1000 ≤ value ≤ 1900
SS!
SS!?

Set slope. This command is used to read the slope calibration parameter for the SF command. This parameter compensates for a variety of factors which affect the magnitude of the coefficient between magnetic coil current and transition frequency. **Example: SS?** might return 1450, the nominal parameter value. This calibration parameter may not be altered by the end user.

The (factory only) SS! command is used to store the current value of the SS parameter to the unit's EEPROM. The SS!? will return the value of the SS parameter which is used on power-up or restart.

GA?
GA{value}   0≤ value ≤ 10
GA!
GA!?

Gain. This command sets the gain parameter in the frequency lock-loop algorithm. Higher gain values have shorter time constants, (the time constant is the time it takes for the frequency lock-loop to remove 67% of the frequency error) but have larger equivalent noise bandwidths (which will reduce the short-term stability of the 10 MHz output.) A gain of 0 will stop the frequency lock-loop so that the frequency of the output is determined by the 10 MHz ovenized oscillator alone. The gain setting, approximate time constants, and approximate equivalent noise bandwidths are detailed in the following table. The gain parameter is set automatically by the program, however, the user may want control over the parameter in special circumstances. **Example: GA7** will set the gain parameter to 7, which has a time constant of about 2 s, which is a typical value for normal operation. **GA?** could return a value of 8 just after restart, which has a short time constant of about 1 s to assist the initial frequency locking. Setting the gain parameter during the first 6 minutes after turn-on or restart will abort the automatic gain sequencing.

Command	Time Constant (seconds)	Noise Bandwidth (Hz)
GA 0	Infinite	0
GA 1	128	0.002
GA 2	64	0.004
GA 3	32	0.008
GA 4	16	0.016
GA 5	8	0.032
GA 6	4	0.064
GA 7	2	0.128
GA 8	1	0.256
GA 9	0.5	0.512
GA 10	0.25	1.024

The **GA!** command stores the current value of the frequency lock loop gain parameter into the unit's EEPROM. **Example:** If the current value of the gain is 6, the command **GA!** will write 6 to the unit's EEPROM which will be used to initialize the gain parameter after the next power-on or restart. Then **GA!?** will return a 6.

PH?
PH{value} 0≤ value ≤ 31
PH!

PH!?

Phase. This command is used to set the phase of the synchronous detection algorithm. The frequency lock-loop (FLL) uses the in-phase component of the photo-signal at the modulation frequency (70 Hz) as the error signal for the FLL. The phase between modulation source and the error signal is affected by phase shifts in the modulation and signal filters and by optical pumping time constants. This parameter corrects for the accumulation of all of these phase shifts. Each modulation cycle consists of 32 phase slots, so each phase increment corresponds to 11.25°. **Example: PH?** would typically return a value of 24.

The PH! command is used to write the current phase parameter into the unit's EEPROM. This is a factory only command. The value which is burned in EEPROM is used on power-on and restart, and may be queried by the **PH!?** command. **Example: PH!?** could return a typical value of 24.

Frequency Synthesizer Control

A frequency synthesizer, which uses the 10 MHz OCXO as a frequency reference, is used to generate the RF which sweeps the rubidium hyperfine transition. The frequency synthesizer multiplies the 10 MHz by a factor M = 19 * (64*N + A) / R, to generate a frequency near 6.834 GHz. (The factor of 19 is from frequency multiplication in the step recovery diode, and the other terms come from the operation of the dual modulus frequency synthesizer integrated circuit.)

The apparent transition frequency is different for each physics package, due mostly to variations in the fill pressure of the resonance cell. The frequency synthesizer parameters, R, N and A, are used to adjust the frequency synthesizer's output frequency to the closest frequency just above the apparent transition frequency, then the magnetic field is set to move the transition frequency up to the synthesizer frequency.

During frequency locking, the frequency of the 10 MHz OCXO is adjusted to maintain the output of the frequency synthesizer on the rubidium hyperfine transition frequency. Initial calibration of the unit will involve finding the synthesizer parameters and magnetic field value which will lock the 10 MHz OCXO at exactly 10 MHz.

During the lifetime of the unit, there will be some aging of the physics package, which will cause the apparent transition frequency to change. This is usually corrected by minor calibration adjustments of the magnetic field strength, which provides a setting resolution of a few parts in 10^{-12}. (See the MO command.) However, if the magnetic field strength reaches its lower or upper limit, it is necessary to change the frequency synthesizer parameters, which can change the output frequency in steps of about one part in 10^{-9}.

The table in Appendix A details the values for R, N and A for the range of frequencies needed.

SP?
SP {R,N,A} $1500\leq R \leq 8191$ $800\leq N \leq 4095$ $0\leq A \leq 63$
SP!
SP!?

Set Parameters. This command is used to set or query the frequency synthesizer's parameters, which will coarsely adjust the unit's output frequency. These parameters may need to be adjusted if the unit cannot be calibrated by magnetic field adjustment.

Example: During calibration, a unit's 10 MHz output frequency is found to be low by 0.010 Hz, and the magnetic field offset adjustment is already at its maximum. (See the MO command.) Sending the **SP?** command returns the current values of R, N and A which are 2610,1466,63 in this example. This corresponds to line 38 in the table in Appendix A. To *increase* the frequency of the 10 MHz output, we select the next *higher* setting, line 37, which will *increase* the frequency by 0.01986 Hz. To do this, we send the command **SP 5363,3014,22** (which are the parameters from line 37). Waiting for the frequency to settle, we now measure the output to be about 0.0098 Hz high. Now the magnetic field is adjusted down to calibrate the unit to exactly 10 MHz. (The **SP!** command is used to save these new values in EEPROM for the next power-on or restart. Also see the MO command for adjusting the magnetic field.)

The **SP!** command is used to write the current frequency synthesizer parameters to the unit's EEPROM for use after the nest restart or power-on cycle. This command is used after the SP command is used during the calibration of the unit. **Example: SP!** will write the frequency synthesizer parameters (R, N and A) which are currently in use to the unit's EEPROM. **SP!?** will return the values for R, N and A which are currently in the unit's EEPROM. The **SP!?** command may be used to verify that the **SP!** write command executed correctly.

Magnetic field Control

A magnetic field coil inside the resonance cell is used to tune the apparent hyperfine transition frequency. The magnetic field is controlled by a 12bit DAC. Increasing the magnetic field will increase the hyperfine transition frequency, which will increase the frequency of the 10 MHz output. The transition frequency may be tuned over about $\pm 3 \times 10^{-9}$ by the magnetic field, which corresponds to ±0.030 Hz at 10 MHz. The output frequency (at 10 MHz) tunes quadratically with field strength, and $\Delta f(\text{Hz}) \approx 0.08 * (\text{DAC}/4096)^{2}$.

A minimum magnetic field should always be present to avoid locking to the wrong Zeeman component of the hyperfine transition, so the 12bit DAC may be set from 1000 to 4095 with 3000 being the nominal midscale value. (A DAC value of 1000 corresponds to about 6% of the full-scale frequency tuning range, 3000 corresponds to about 53%, while 4095 is 100% of the full-scale range.)

To help cancel frequency shifts due to external magnetic fields, the current in the coil is switched at a 5 Hz rate. The frequency lock-loop averages over a full period of the switch rate to avoid injecting a spur at 5 Hz onto the 10 MHz control signal. The switching of the magnetic field is enabled at power-on and restart, but may be turned on or off by RS232 command. (see MS command.)

The commands associated with magnetic field control (MO, MS, and MR) allow direct control of the magnetic field circuitry. Most users will not want to control the magnetic field directly, but will instead allow the program to read the frequency calibration pot or external control voltage and then control the magnetic field. If they want software control of the unit's calibration, they may choose to use the SF commands, which disable the analog control and allow the frequency to be adjusted over a range of $\pm 2000 \times 10^{-12}$. (The program will linearize the magnetic field control of the frequency offset with either analog or software control.)

MS?
MS{0 or 1}

Magnetic switching. The MS command is used to turn off or on the 5Hz switching of the frequency tuning magnetic field. Magnetic switching is enabled when the unit is powered-on or after a restart. (Since the PRS10 is calibrated with the field switching enabled, turning off the field switching may alter the calibration.) **Example:MS 1** will turn on the magnetic field switching, and **MS 0** will turn it off. **MS?** will return a "1" if the field switching is currently enabled.

MO?
MO{value} 2300 ≤ value ≤ 3600
MO!
MO!?

Magnetic offset. The magnetic offset is the value, determined when the unit is calibrated, which calibrates the unit to 10 MHz. The restricted range is necessary to allow room for user calibration via the internal frequency calibration pot or by an external voltage. If the unit cannot be calibrated to 10 MHz within the allowed range of MO values, then a different setting for the frequency synthesizer is required. (See SP command and the table in Appendix A. **Example: MO 3000** sets the magnetic offset to 3000, which is its nominal (mid-linear scale) value. The **MO?** command reads back the current value of the magnetic offset. **MO!** is used to store the current value of the magnetic offset parameter to EEPROM for use after the next restart. **MO!?** may be used to query the value stored in EEPROM. This value is used on power-up or restart.

MR?

Magnetic read. This command returns the value that the 12bit DAC is using to control the magnetic field. This value is computed from the magnetic offset value (see MO command) and the position of the internal frequency calibration pot, external calibration voltage, or value sent by the SF command.

The value is computed from the equation DAC = $\sqrt{(SF*SLOPE + MO^2)}$ where SF is the desired frequency offset in parts per 10^{-12} (from the cal pot position, the SF command, or the 1 pps PLL and is in the range –2000 < SF < 2000), SLOPE is the SF calibration factor with a nominal value of 1450 (see SS command), and MO is the magnetic offset value. The returned value should be in the range of 1000 to 4095.

Example: MR? would return a value of 3450 if the magnetic offset is at 3000, the SF command requested an offset of +2000 x 10^{-12}, and the SS CAL factor has the nominal value of 1450.

Frequency Control

The frequency of the 10 MHz output may be adjusted in a number of ways: the internal calibration potentiometer may be set (accessible via a hole in the bottom plate), an external voltage (0 to +5.00 Vdc, applied to the interface connector pin 2) can override the internal pot, or, these analog channels may be overridden with a software command which sets the frequency directly.

When the unit turns on, or after a restart command, the control program will default to reading the analog channel for frequency calibration. (This is important to maintain compatibility with existing sockets.) The calibration pot and the external voltage control provide a full-scale tuning range of ± 2000 x 10^{-12}, with a worst case resolution of 5 x 10^{-12}.

All of the channels for calibrating the unit are linearized, so that the frequency characteristic will be linear with applied voltage, pot setting, or SF value even though the transition frequency changes quadratically with field strength.

One pulse per second (1 pps) control

To facilitate system integration, the PRS10 provides a 1 pps output which may be set over an interval from 0 to 999,999,999 ns with 1 ns resolution. The unit also has the ability to measure the arrival time of a 1 pps input over the same interval and with the same resolution.

The ability to time-tag a 1 pps input allows the PRS10 to be phase-locked to other clock sources (such as the 1 pps output from a GPS receiver) with very long time-constants.

This is a very useful feature for network synchronization, and allows the configuration of a reliable Stratum I source at a very low cost.

TT?

Time-tag. This command returns the value of the most recent time-tag result in units of nanoseconds. If a new time-tag value is not available then -1 (the only case for which the returned value is negative) will be returned. **Example: TT?** would return the value 123456789 to indicate that the most recent 1 pps input arrived 123,456,789 ns after the 1 pps output. Returned values range from 0 to 999999999.

TS?
TS{value}   7000 ≤ value ≤ 25000
TS!
TS!?

Time slope. This command is used to calibrate the analog portion of the time-tagging circuit. The analog portion is used to digitize the time of arrival with 1 ns resolution and 400 ns full-scale. (Counters are used for the portion of a time interval longer than 400 ns.) The analog circuit stretches the time interval between the 1 pps input and the next edge of a internal 2.5 MHz clock by a factor of about 2000, and measures the duration of the stretched pulse by counting a 2.5 MHz clock. The analog portion of the time-tag result is calculated from the equation $\Delta T(ns) = counts * TS / 2^{16}$, where TS is the time slope value, which has a nominal value of 13,107.

Example: TS? might return 14,158 which is a time slope parameter value a bit above the nominal value, which would be required if the analog portion of the time-tagging circuit stretched the pulse by a bit less than a factor of 2000. **TS?** will return the current value of the time slope.

The TS! command is used to write the current value of the time slope parameter into the unit's EEPROM. The TS {value} and TS! are factory only commands. Example: TS! will write the current value of the time slope (which may be queried with the TS? command) to the unit's EEPROM. **TS!?** will return the time slope calibration factor which is in the unit's EEPROM.

TO?
TO{value}   -32767 ≤ value ≤ 32768
TO!
TO!?

Time offset. This calibration value (in ns) is subtracted from the measured time-tag value to align the result with the 1pps output. To calibrate, the 1pps output is connected to the 1pps input and the result is read with the TT? command. The returned value is sent with the TO command to calibrate the offset.

Example: Suppose that the 1pps output is connected to the 1pps input. The time tag value is read with the TT? query, and returns a value of 125. The command TO 125 is sent to correct for the offset. After waiting about one second (to allow another time-tag value to be acquired) the next TT? query returns a value of 3 (indicating a measurement of 3 ns after the 1 pps output) Waiting another second, the next TT? query returns the value 999999998 (indicating 2 ns before the 1 pps output). These values are consistent with a well calibrated time-tag offset.

The TO! command is used to write the current value of the time offset to the unit's EEPROM. Example: TO! will write the current time-tag offset value to the unit's EEPROM for use after the next power-up cycle or restart command. **TO!?** will return the value which is burned in the unit's EEPROM. The TO{value} and TO! commands are factory-only commands.

PP?
PP{value} 0 ≤ value ≤ 999999999

Place pulse. This command is used to move the 1 pps output from its current position. The 1 pps output can be moved earlier in time by 1 ns to 999999999 ns. Since the 1 pps input time-tag is referenced to the 1 pps output, changing the 1 pps output placement will change the report time-tag values as well. (See the TT and TO commands.) **Example: PP 123456789** will move the 1 pps pulse train earlier by 123,456,789 ns.

PS?
PS{value} 100 ≤ value ≤ 255
PS!
PS!?

Pulse slope calibration. This command is used to calibrate the analog portion of the 1 pps output time delay circuit. This circuit is used to delay the 1 pps pulse train with 1 ns resolution and 100 ns full-scale. (Counting logic is used for the portion of the time interval longer than 100 ns.) The pulse slope value corresponds to the DAC8 value which provides a delay closest to (but not exceeding) 100 ns. Example: PS 200 set the pulse slope to its nominal value of 200. (PS{value} is a factory only command.) The **PS?** command will return the current value of the pulse slope.

The PS! command writes the current value of the pulse slope to the unit's EEPROM for use after the next power-on or restart. This command is used after the pulse output analog output is calibrated. Example: PS! will write the current value of the pulse slope (which calibrates the 100 ns analog delay portion of the 1 pps pulse delay circuit) to the unit's EEPROM. Note that PS! is a factory-only command.

1PPS Locking Control

To facilitate integration into systems which require very low aging, automatic calibration, or a traceable frequency standard, the PRS10 may be locked to an external 1 pps input.

A second-order digital PLL is used to lock the unit's frequency (both the 10 MHz and 1 pps outputs) to an external 1 pps input with time constants ranging from 256 s to 65536s (about 4 minutes to about 18 hours).

When provided with an accurate and stable 1 pps source, the unit will automatically align its 1 pps output to the 1 pps input and then adjust the frequency of the rubidium reference to maintain the alignment over time. A typical application would lock the PRS10 to the 1 pps output from a GPS receiver with a time constant of several hours.

Several commands and one status byte may be used to control and monitor the PLL, however, default values will allow units to lock to clean 1 pps inputs without any software interaction.

PL?
PL{0 or 1}
PL!
PL!?

Phase lock control. This command may be used to disable the 1 pps PLL, or to re-enable (and so restart) the 1 pps PLL. The unit is shipped with the phase lock control enabled. This command would be used if the 1 pps time-tagging were being used to measure the position of 1 pps inputs and phase locking is not desired. **Example: PL 0** will disable the PLL to the 1 pps inputs so that the frequency of the rubidium standard will not be affected by the 1 pps inputs. **PL?** will return a "1" if the PLL to the 1 pps is enabled. **PL!** is used to write the current value (0 or 1) to the EEPROM for use after the next start up. **PL!?** is used to query the value of the phase lock control parameter which is stored in the unit's EEPROM.

PT?
PT{value} $0 \leq \text{value} \leq 14$; $\tau_1 = 2^{(\text{value}+8)}$ seconds (256, 512, ... 4,194,304)
PT!
PT!?

Phase-lock time constant. This command is used to set the integrator's time constant, τ_1, of the 1 pps PLL. The time constant is equal to $2^{(\text{value}+8)}$ seconds. The default value is 8, which provides a time constant of $2^{(8+8)}$ or 65536 seconds. Integrator's time constants can range from 256 to 4,194,304 seconds.

Example: PT 14 sets the time constant to the longest value, which is a little more than 18 hours (see the following table). This would provide the most smoothing (the least equivalent noise bandwidth) but would reduce the ability of the PLL to correct against abrupt ambient changes which might affect the PRS10's output frequency. **PT?** will return the current value of the time constant parameter. A phase lock time constant may be stored in EEPROM as a new default with the **PT!** command. The **PT!?** command may be used to verify the value stored in EEPROM.

Command	Time Constant	Natural Time Constant
PT 0	4.2 min	8.4 min
PT 1	8.5 min	11.9 min
PT 2	17.1 min	16.9 min
PT 3	34.1 min	23.8 min
PT 4	1.1 hrs	33.7 min
PT 5	2.3 hrs	47.7 min
PT 6	4.6 hrs	1.1 hrs
PT 7	9.1 hrs	1.6 hrs
PT 8	18.2 hrs	2.2 hrs
PT 9	1.5 days	3.2 hrs
PT 10	3.0 days	4.5 hrs
PT 11	6.1 days	6.4 hrs
PT 12	12.1 days	9.0 hrs
PT13	24.3 days	12.7 hrs
PT 14	48.5 days	18.0 hrs

PF?
PF{value} $0 \leq$ value ≤ 4 ; (value:ζ) : (0:1/4, 1:1/2, 2:1, 3:2, or 4:4)
PF!
PF!?

Phase-lock stability factor. This command is used to set the stability factor, ζ, of the 1 pps PLL. The stability factor is equal to $2^{(value-2)}$. The default value is 2, which provides a stability factor of $2^{(2-2)} = 2^0 = 1$. Stability factors can range from 0.25 to 4.0. **Example: PF 1** sets the stability factor to 0.5, which will reduce the equivalent noise bandwidth of the PLL at the cost of increasing the ringing near the natural frequency (relative to the default settings). **PF?** will return the current value of the stability factor parameter. **PF!** may be used to write the current stability factor to the EEPROM for use as the new default. **PF!?** may be used to read the value of the stability factor which is stored in EEPROM.

PI?
PI {value} $-2000 \leq$ value ≤ 2000

Phase-lock integrator. This command is used to set the value of the integral term in the PLL's digital filter. It is not necessary to set this value, as it will be initialized by the PLL routine to the current frequency setting parameter when the PLL begins. Users may want access to the value to alter the PLL characteristics, or to investigate its operation. **Example: PI 0** will set the integrator in the PLL's digital filter to 0, which is the center of the ±2000 bit range. **PI?** will return the current value of the PLL integrator. (There are two terms which control the phase locking of the PRS10 to an external 1pps source: the integral term and the proportional term. The proportional term is equal to the value returned by an **SF?** minus the value returned by the **PI?**.)

Analog Control

SD{port}?
SD{port,value} 0 ≤ port ≤ 7 and 0 ≤ value ≤ 255 (factory only)
SD{port}!
SD{port}!?

Set DAC. This command is used to set (or read the settings of) an octal 8bit DAC which provides analog signals to control systems parameters. The command which sets values is only available to the factory. The command to query values may be used by all. The query command returns a single integer in the range of 0 to 255.

Port	Function
0	Controls the amplitude of the RF to multiplier in resonance cell
1	Controls the analog portion (0 to 99 ns) of the delay for the 1 pps output
2	Controls the drain voltage for the discharge lamp's FET oscillator
3	Controls the temperature of the discharge lamp
4	Controls the temperature of the 10 MHz SC-cut crystal
5	Controls the temperature of the resonance cell
6	Controls the amplitude of the 10 MHz oscillator
7	Controls the peak deviation for the RF phase modulation

Example: SD2? could return the value 255 indicating that the unit has set the discharge lamp's FET drain voltage to the maximum (which it does while it is trying to start the lamp.)

The SD{port}! is a factory only command which writes the data from the corresponding SD port to the unit's EEPROM for use on subsequent restarts. **Example: SD3!?** will return the start-up value for SD3 (lamp temperature control value) which is stored in the unit’s EEPROM.

Analog Test Voltages

AD{port}? port = 0,1,2,...15

Analog to digital. This command reads the voltage at the corresponding 12bit ADC port and returns the voltage as a floating point number. Values can range from 0.000 to 4.998. The voltages correspond to various test points in the system per the following table. Note that this command can only query. **Examples: AD10?** could return the value 0.710 indicating that the case temperature sensor is at 71 °C (this sensor indicates a temperature which is about midway between the baseplate temperature and the lamp temperature.)

Command	Returned voltage
AD 0?	Spare (J204)
AD 1?	+24V(heater supply) divided by 10.
AD 2?	+24V(electronics supply) divided by 10
AD 3?	Drain voltage to lamp FET divided by 10
AD 4?	Gate voltage to lamp FET divided by 10
AD 5?	Crystal heater control voltage
AD 6?	Resonance cell heater control voltage
AD 7?	Discharge lamp heater control voltage
AD 8?	Amplified ac photosignal
AD 9?	Photocell’s I/V converter voltage divided by 4
AD 10?	Case temperature (10 mV/°C)
AD 11?	Crystal thermistors
AD 12?	Cell thermistors
AD 13?	Lamp thermistors
AD 14?	Frequency calibration pot / external calibration voltage
AD 15?	Analog ground

AD{port}? 16 ≤ port ≤ 19

A/D via CPU’s E-port. This command returns a value corresponding to the voltage present at the input to the microcontroller’s octal 8bit ADC (port E on the MC68HC11). Only the first four ports are in use. The voltage corresponds to various test point in the system per the following table. **Example: AD17?** could return a value of 4.81 indicating that the 360 MHz RF synthesizer has acquired lock.

Command	Returned voltage

AD 16?	Varactor voltage for 22.48 MHz VCXO (inside RF synthesizer) / 4
AD 17?	Varactor voltage for 360 MHz VCO (output of RF synthesizer) / 4
AD 18?	Gain control voltage for amplifier which drives frequency multiplier / 4
AD 19?	RF synthesizer's lock indicator voltage (nominally 4.8 V when locked)

Status Bytes

ST?

Status query. This command returns the six system status bytes which are used to indicate the health and status of the unit. The values ranges from 0 to 255. The six status bytes are detailed in the tables below. A status bit will remained set until it is read, even though the condition which caused the error has been removed. Some status bits are not errors: for example, during warmup the status bytes may indicate that the lamp is not lit, temperatures are low, and the unit is not locked.

Example: Immediately after power is applied to a unit, the command **ST?** returns 32, 2, 42, 1, 2, 129. From the status byte definitions below, we see that the following conditions exist:

32… the lamp has not yet started
2… the RF VCXO has not yet locked
42… the lamp, crystal, and cells are all below their set point temperatures
1… the frequency lock has not been established
2… fewer than 256 1pps inputs have been qualified
129.. both the lamp and unit have been restarted

ST1 : Power supplies and Discharge Lamp

ST1 bit	Condition which sets bit	Corrective Action
0	+24 for electronic < +22 Vdc	Increase supply voltage
1	+24 for electronics > +30 Vdc	Decrease supply voltage
2	+24 for heaters <+22 Vdc	Increase supply voltage
3	+24 for heaters > +30 Vdc	Decrease supply voltage
4	Lamp light level too low	Wait: check SD2 setting
5	Lamp light level too high	Check SD2 setting
6	Gate voltage too low	Wait: check SD2 setting
7	Gate voltage too high	Check SD2 setting

ST2: RF Synthesizer

ST2 bit	Condition which sets bit	Corrective Action
0	RF synthesizer PLL unlocked	Query SP? verify values
1	RF crystal varactor too low	Query SP? verify values
2	RF crystal varactor too high	Query SP? verify values
3	RF VCO control too low	Query SP? verify values
4	RF VCO control too high	Query SP? verify values
5	RF AGC control too low	Check SD0? values
6	RF AGC control too high	Check SD0? values
7	Bad PLL parameter	Query SP? verify values

ST3: Temperature Controllers

ST3 bit	Condition which sets bit	Corrective Action
0	Lamp temp below set point	Wait for warm-up
1	Lamp temp above set point	Check SD3, ambient
2	Crystal temp below set point	Wait for warm-up
3	Crystal temp above set point	Check SD4, ambient
4	Cell temp below set point	Wait for warm-up
5	Cell temp above set point	Check SD5, ambient
6	Case temperature too low	Wait for warm-up
7	Case temperature too high	Reduce ambient

ST4: Frequency Lock-Loop Control

ST4 bit	Condition which sets bit	Corrective Action
0	Frequency lock control is off	Wait for warm-up
1	Frequency lock is disabled	Enable w/LO1 command
2	10 MHz EFC is too high	SD4,SP,10MHz cal,Tamb
3	10 MHz EFC is too low	SP, 10 MHz cal
4	Analog cal voltage > 4.9 V	Int cal. pot, ext cal. volt
5	Analog cal voltage < 0.1	Int cal. pot, ext cal. volt
6		
7		

ST5: Frequency Lock to External 1pps

ST5 bit	Condition which sets bit	Corrective Action
0	PLL disabled	Send PL 1 to enable
1	< 256 good 1 pps inputs	Provide stable 1 pps inputs
2	PLL active	
3	> 256 bad 1 pps inputs	Provide stable 1 pps inputs
4	Excessive time interval	Provide accurate 1 pps
5	PLL restarted	Provide stable 1 pps inputs
6	f control saturated	Wait, check 1 pps inputs
7	No 1 pps input	Provide 1 pps input

ST6: System Level Events

ST6 bit	Condition which sets bit
0	Lamp restart
1	Watchdog time-out and reset
2	Bad interrupt vector
3	EEPROM write failure
4	EEPROM data corruption
5	Bad command syntax
6	Bad command parameter
7	Unit has been reset

Calibration Procedures

Many applications for the PRS10 only require that the frequency of the 10 MHz output be calibrated. This may be done by adjusting a potentiometer, which is accessible through a hole in the bottom of the unit. The unit should be operating for at least 24 hours before it is calibrated. The 15 turn pot has a range of ±0.020 Hz. The frequency increases if the pot is turned counter-clockwise, by about 0.001 Hz for 3/8's of a turn.

Note: the potentiometer position will not affect the frequency of operation if : (1) it is turned to either extreme, (2) an external control voltage is applied to pin 2 of the main connector, J100, (3) an SF (set frequency) command has been sent via the RS232 interface, or, (4) the unit is locked to an external 1pps input. The time constant for pot adjustments depend on the setting of the frequency lock-loop gain (see GA command: the default is about 2 seconds.)

In the case that the unit cannot be calibrated because the internal pot has reached an extreme position, it will be necessary to modify a calibration values which are stored in the unit's EEPROM. To verify that the pot has been turned to a limit of its motion, measure the voltage on pin 2 (POT_W) of J100 (the main connector) with respect to the chassis. Zero volts on pin 2 indicates that the pot has been adjusted for the lowest frequency, and +5.0 Vdc indicates that the pot has been adjusted for the highest frequency.

To modify EEPROM calibration values, it will be necessary to establish RS232 communications with the PRS10 DUT. This can be done with a three wire connection between the PC COM: port and the PRS10's main connector. A communication program (see Windows Accessories or other) will be needed as well. (See MO and SP commands.)

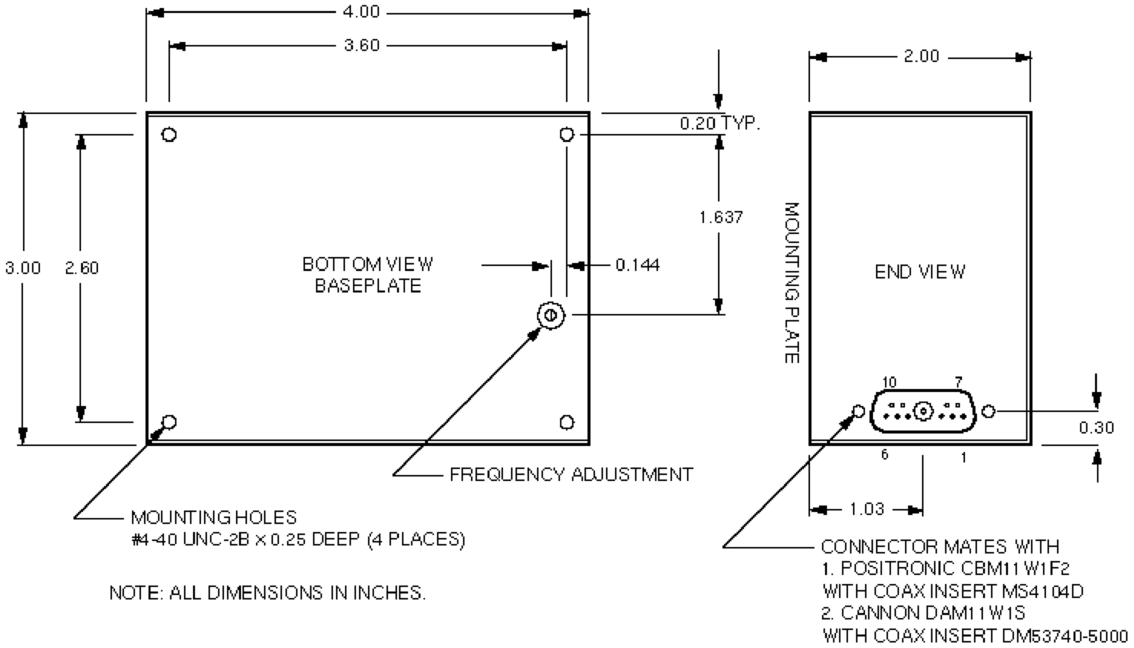

Figure 4. Mechanical Dimensions