100KHz MULTICHANNEL PWMCTA Operational Manual

Digital Flow Technologies, Inc.
2353 Sapphire Lane
East Lansing, MI 48823
May 2, 2003

Index

5. Reregularization – taureg.m
6. x-wire processing – pwmxprox.m

THE PWM-CTA INSTRUCTION MANUAL

Nomenclature

A,B,n	Calibration coefficients; see (3.9)
C-CTA	Conventional Constant Temperature Anemometer
$E_A, E_B, E_{B'}, E_{Ref}$	Voltages at important points in the circuit; see Fig. 2 and text material, section 1.2, for clarification
OHR	Overheat ratio for the sensor
R_h, R_c, R_s	The maximum (i.e., the "hot") resistance of the sensor, the cold resistance of the sensor, the sensor resistance at a given instant of time
τ_j	Duration of the sensor heating for the jth period
$<T_j>$	The time period between equal energy states for which τ_j is the duration of the sensor heating
$<V_j>$	The temporally averaged cooling velocity in the period $<T_j>$

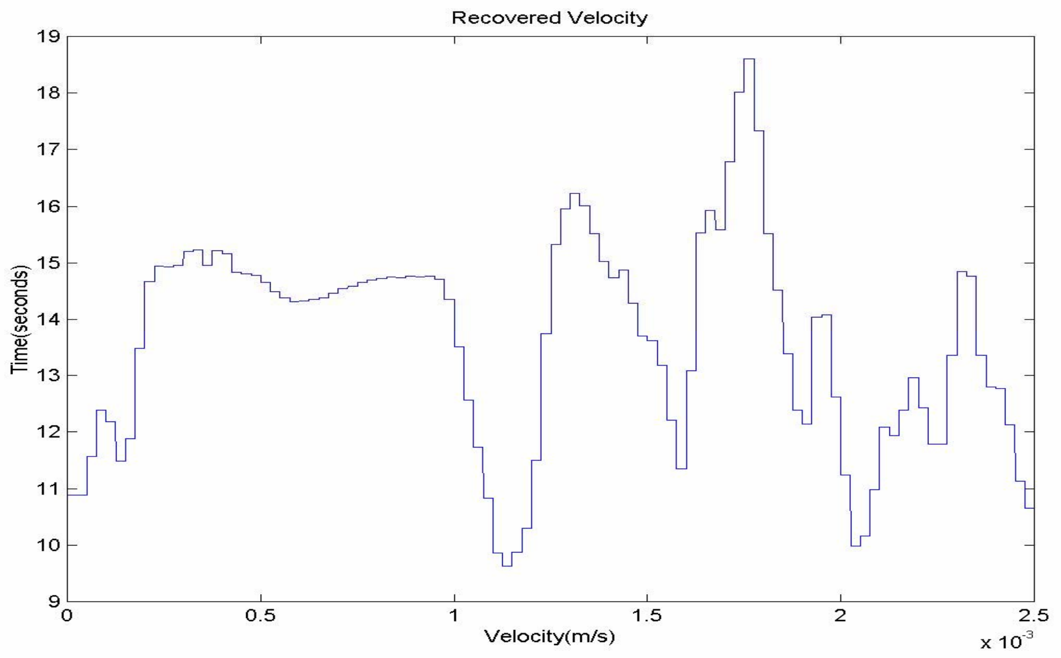
1. Principle of Operation

1.1. Introduction

The pulse width modulated-constant temperature anemometer (PWM-CTA) represents a significant departure from the conventional-CTA (C-CTA) that has been commercially available since the 1960's. The latter device acts as a servo system whose output voltage (E_{Bridge}) seeks to faithfully reflect the (time) continuous function: the velocity, at the sensor. (A single sensor would seek to resolve the magnitude normal to itself (V), the two sensors of an x-array, the in-plane magnitude and direction, etc.)

In contrast, the PWM-CTA seeks to identify the temporal average magnitude(s) for a very short time segment (T) of the continuous velocity. Specifically, the experimental data of Fig. 1 shows the PWM-CTA output, from a single sensor, at the $\bar{u}/U_o \cong 0.68$ location of a planar jet. The figure (in part (b)) also shows a "continuous" V(t) record – that was created from the stepped response of part (a) – by connecting the midpoints of the record. The latter is, of course, similar to the representation that one achieves by straight line connections between the processed "instantaneous" samples that have been captured by an A/D converter and post-processed: $E(t) \rightarrow V(t)$. This figure is to emphasize, in its contrast between parts (a) and (b), the similarities and differences between representing V(t) using discrete samples of the continuous record (represented in part (b)) and the short-term (T) averages of V(t) (part (a)).

Note: To the user of this 30 April, 2003 version of the Instruction Manual. The calibration data for an X-array and the schematic showing the evaluation of (Q,γ) from τ_1/T and τ_2/T are not currently available. These will be represented by Figs. ? and ? in the revised manual. This information will be forwarded when available. Contact <foss@egr.msu.edu> for updates.



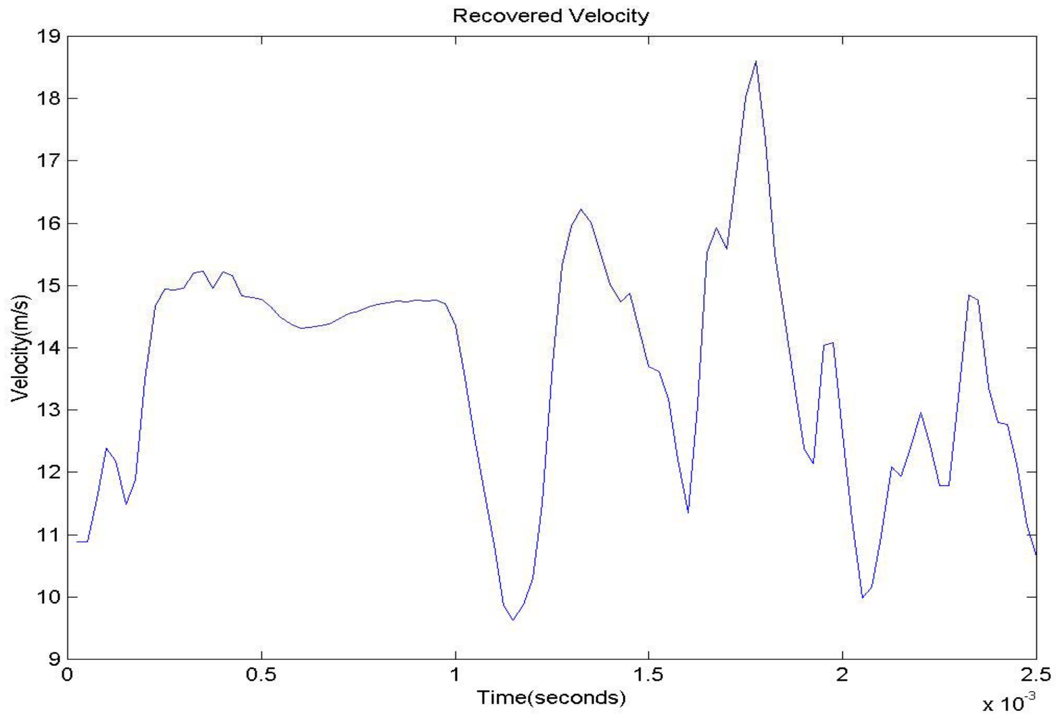


Figure 1: Velocity time series: $\overline{u}/U_o \cong 0.68$ in a planar jet
a) PWM-CTA output
b) Simulated u(t)…same data

1.2. The Basic Circuit – Simplified

The electronics that provide the true average of V(t) for the time period

$$t_j \leq t \leq t_j + T \qquad (1.2.1)$$

are shown schematically in Fig. 2.

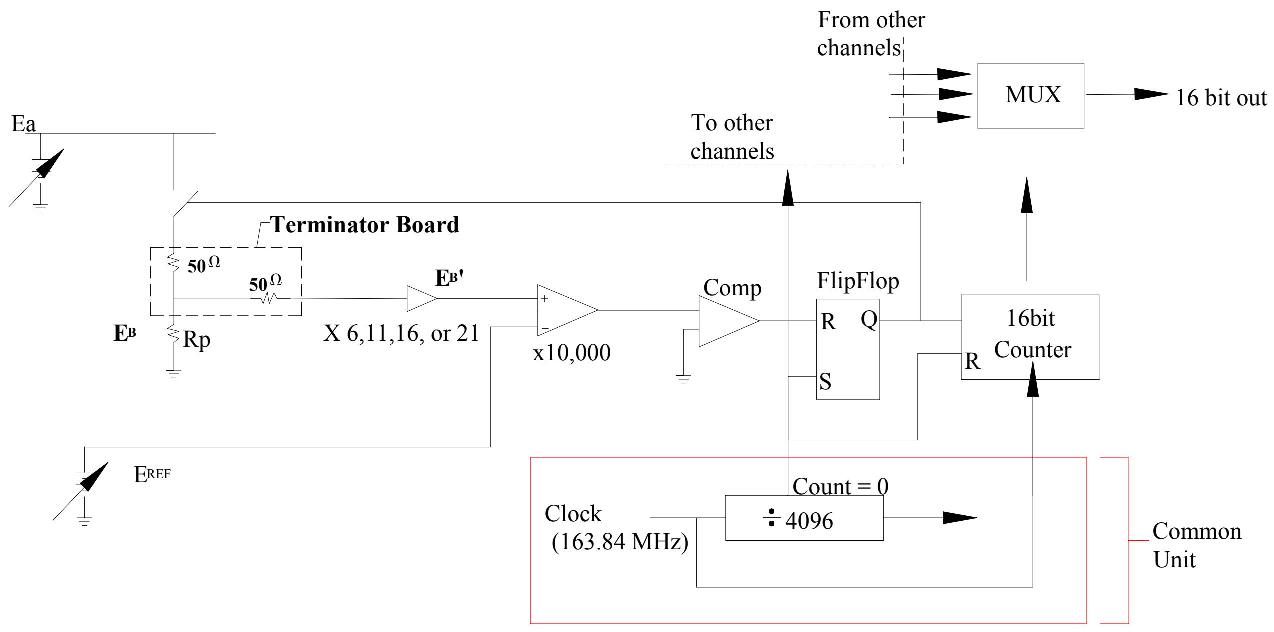


Figure 2: Simplified PWM-CTA Schematic
Note: R_1=50Ω for a typical (R_C=3.5Ω) hot-wire sensor

The essential operational feature is that the voltage: E_A, is imposed on the series resistor R_1 plus the sensor R_s for a period that is sufficient to bring the sensor temperature – and resistance – to the preset values: T_H, R_H. (Given the linearity between the sensor temperature and its resistance, the OHR may be defined using the resistance as: $R_H = (1 + OHR)R_c$ where OHR = overheat ratio and R_c = cold sensor resistance).

1.1. The Circuit's Operation

1.1.1. The Sensor's Response to V(t)

The time dependent sensor resistance, $R_s(t)$, can be inferred from the output of the PWM-CTA. A short time segment from the data set, which was used to produce <V(t)> in Fig. 1, is presented in Fig. 3. (The 5μ diameter tungsten sensor was operated with an OHR of 0.7).

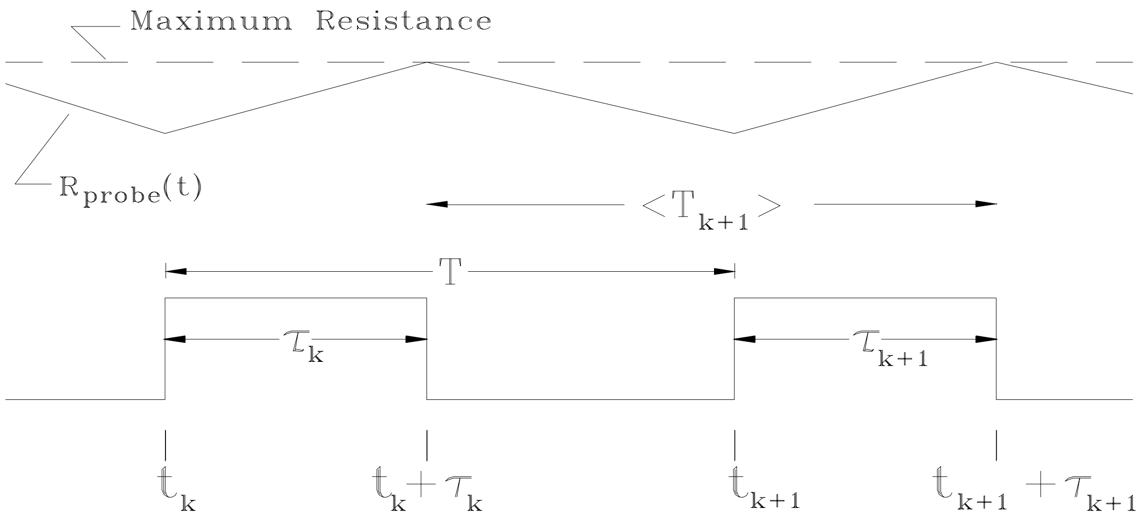


Figure 3: Timing Diagram for $R_w(t)$

As shown in Fig. 3, the sensor is heated during the τ periods and allowed to cool for the balance of the T cycle.

<u>Important Note</u>: The operator will set the adjustable controls (see Section 5.3) such that

$$\tau_{maximum} \leq T/2 \tag{1.3.1}$$

The operator will also check to ensure that

$$\tau_{minimum} \approx T/4 \tag{1.3.2}$$

The latter condition ensures that the intrinsic switching noise – associated with the start of the heating cycle at t_j – does not interfere with the sensitive trigger that is activated when

$$E_{B'} = E_{Ref'} \tag{1.3.3}$$

at time $t_j+\tau_j$ as shown in Fig. 2. The condition for τ_{max} ensures that an adequate cooling period ($t_j+\tau_j<t<t_{j+1}$) is included in the operating cycle.

There are two "free parameters" which are evident in Figs. 2 and 3; namely, E_A and T. Their purpose of adjustments is as follows:

i) select T to establish the desired sample frequency,

ii) set the E_A value, when the sensor is exposed to the maximum velocity, to ensure that $\tau_{max}/T \leq 0.5$.

The operator's actions, which establish the correct operating conditions, are made explicit in Section 7.20.2.3. Note that: if τ_{min} is too small, false triggering may occur. If τ_{min} is too large, insufficient resolution will occur. It may be necessary to adjust T to a higher or lower value to avoid these problems. This is explicitly considered in Section 7.20.2.3.

1.3.2 The Transfer Function: $\tau \rightarrow V$, for the PWM-CTA

The essential nature of the PWM-CTA is shown in Fig. 3; namely, that the $R_s(t)$ value is brought to R_H at the time values: $t_j + \tau_j$. Since a given R_H value denotes a given energy state in the sensor, one can precisely relate the input energy:

$$\text{Energy Input} = \int_{t_j}^{t_j+\tau_j} \frac{E_B^2(t)}{R_s(t)}\,dt \tag{1.3.4}$$

to the energy transferred from the sensor in the period identified in (1.3.5). Namely,

$$\text{Energy Transferred: } \int_{t_{j-1}+\tau_{j-1}}^{t_j+\tau_j} [f\,(V)]\,dt\,. \tag{1.3.5}$$

Note that the relationship between these two integrals will be established by a conventional calibration procedure. For example, one might elect to use

$$f(V) = A' + B'V^n \tag{1.3.6}$$

for the rhs function in (3.5). In this case the calibration information (acquired for the conditions of a set of nominally steady state V values) would be

$$[E_A^2/R_H]\tau_j = [A + BV^n][(t_j + \tau_j)] - [(t_{j-1} + \tau_{j-1})]\,. \tag{1.3.7}$$

Note that the constant value: (E_A^2/R_H) has been substituted for the time dependent (E_B^2/R_s) ratio in the integral of (1.3.4). This substitution is satisfactory for two reasons:

i) the minimum value of R_s is of order 0.99 R_H, and

ii) the substitution effects will be accounted for in the calibration coefficients of (3.7).

1.3.3 The Calibration Transfer Function

Equation (3.7) can be written as

$$\frac{\tau}{T} = A + B < V^n > \ldots\text{for the calibration environment}\ldots\frac{\partial V}{\partial t} = 0. \tag{1.3.8}$$

<V> and τ are the steady state values averaged over the calibration "dwell period" and the similarly averaged $[(t_j+\tau_j)-(t_{j-1}+\tau_{j-1})]$ values are equal to the duty cycle: T. Hence, the coefficients A,B,n (or their equivalents) can be determined from a calibration procedure that is identical to that used for the C-CTA. A representative calibration data set for a single sensor is shown in Fig. 4.

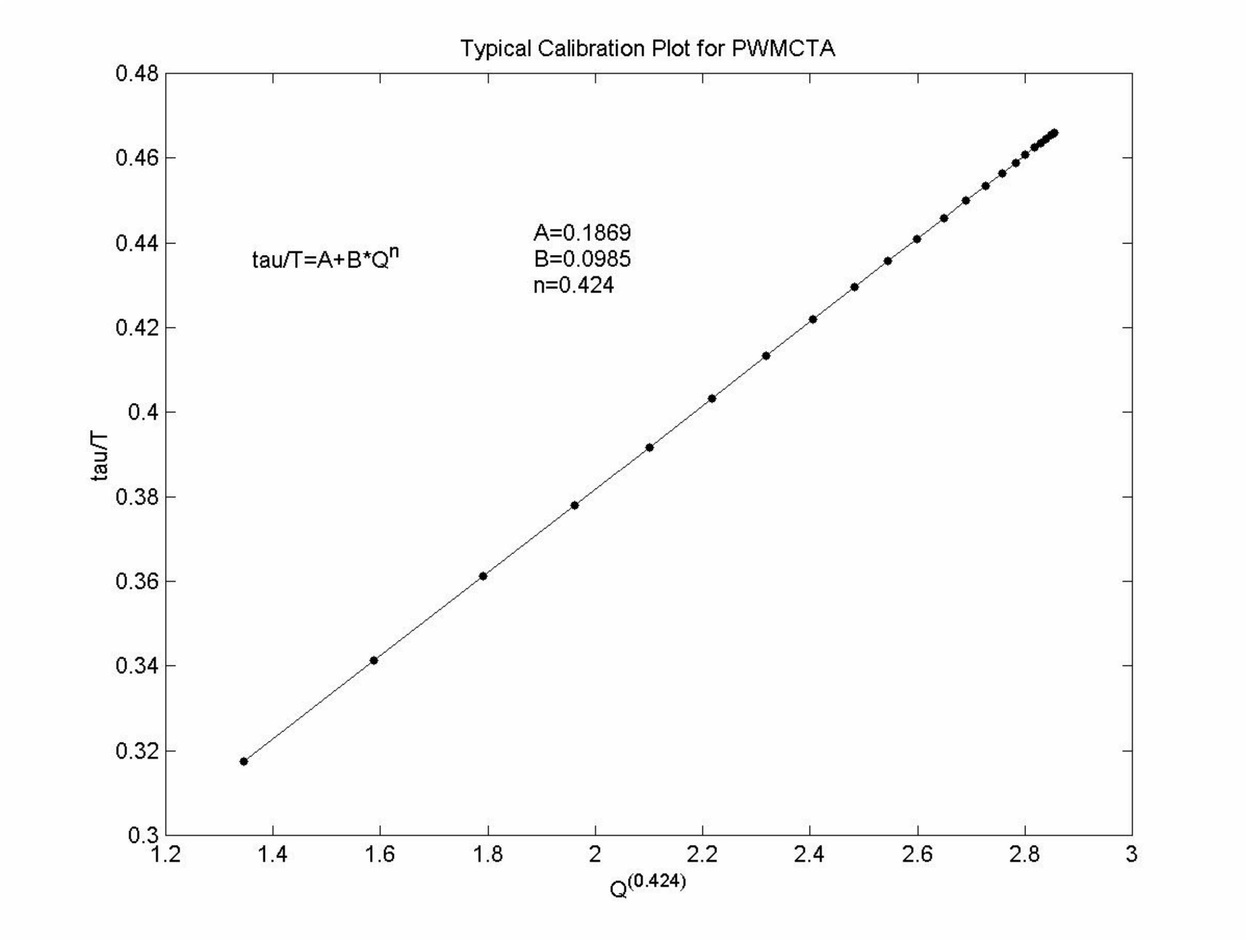


Figure 4: A representative τ/T=f(Q) calibration for a single sensor.

1.3.4 The Operational Transfer Function

The coefficients from (1.3.7) apply to flow field measurements. However, since the velocity is no longer steady, the integration time for the heat transfer process is also a variable. Hence, the generalized transfer function is

$$\frac{\tau_j}{< T_j >} = A + B < V_j >^n \tag{1.3.9}$$

where

$$< T_j > = (t_j + \tau_j) - (t_{j-1} + \tau_{j-1}) \tag{1.3.10}$$

and

$<V_j>$ = the average cooling velocity over the time period of $< T_j >$.
(1.3.11)

1.3.5 The Recovery of a Uniformly "Sampled" Time Series for $<V(t_j)>$

All statistical properties of a turbulent flow, that do not represent frequency content or time dependent information, can be obtained directly from (1.3.9).

If temporally coordinated, multi-sensor measurements (e.g., an x-array, or a two-point correlation function, etc.) are to be made, then it will be important to register the inferred velocity values to a common time base. This common time base also expedites the evaluation of the 1-D energy spectrum representation of the time series data. The following algorithm is one method of achieving this common and regularized time base.

Given that the τ values will be bracketed as

$$\tau_{\min} \approx \frac{T}{4} \text{ and } \tau_{\max} \approx \frac{T}{2},$$

The following regularization algorithm is recommended.[*] Consider a regular time series for which

$$t_{\text{center}} = t_j + \frac{7T}{8} \qquad (1.3.12)$$

is the center point of the averaged quantity to be measured. The implementation of this algorithm is presented in the following sub-sections.

1.3.5.1 An Algorithm for a Single Sensor

[*] The user may wish to organize the regularized data to be centered on the time mean value for the particular record. This can be readily accomplished by substituting $\overline{\tau} + 0.5\,T$ as the center of the regularized record.

The appropriately registered single sensor output (velocity, shear stress, etc.) will be characterized for the $t_j \rightarrow t_{j+1}$ time period. This value will be "assigned" to

$$t_j + \frac{7T}{8}$$

as noted above. As such, it will collect its inputs from the time interval

$$t_j + \frac{3}{8}T \le t < t_{j+1} + \frac{3}{8}T.$$

The following set of conditions represent the possible contributions to the regularized output for the time interval $t_j+(3/8)T$ to $t_{j+1}+(3/8)T$. Note that this is displayed in graphical form where a and b are mutually exclusive as are c and d. The condition c, however, can combine with either a or b; the same is true for d.

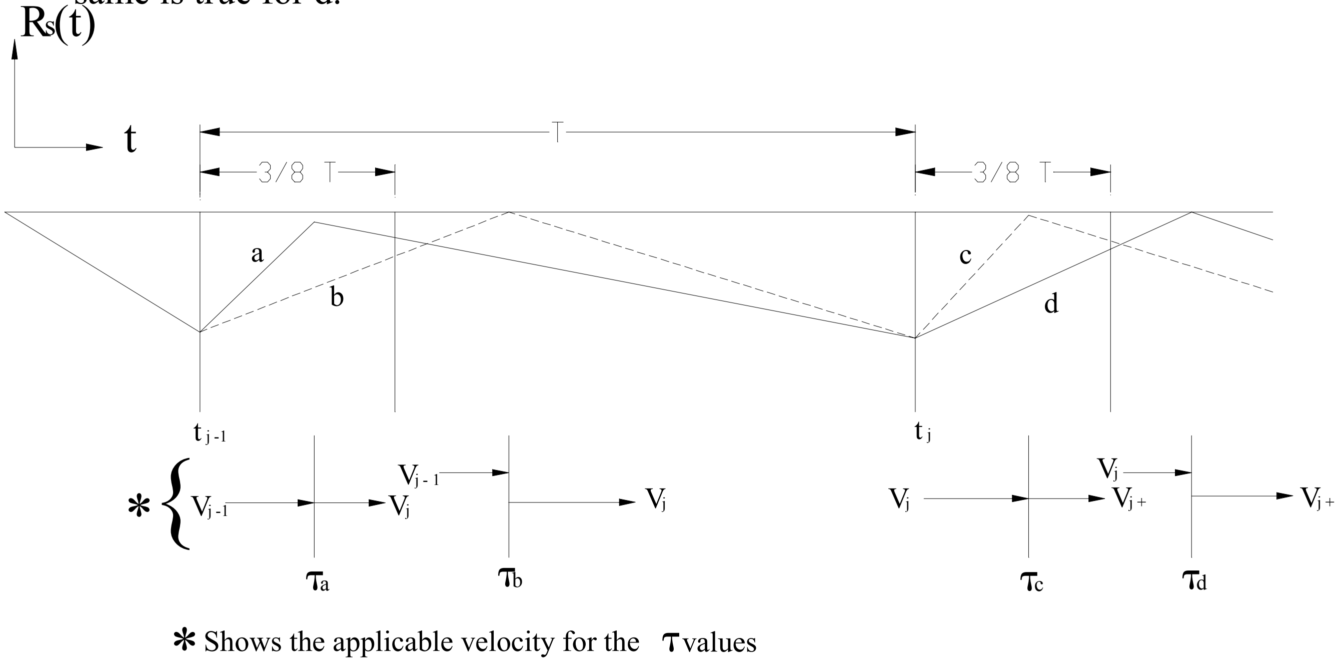


Figure 5: A representation of the algorithm which creates a regularized <V(t_j)> time series.

The contributions to the regularized velocity magnitude in the above period can be expressed using the a…d code as

if a, $<V_j>\left[\frac{5}{8}T\right]$

if b, contribution as

$$< V_{j-1} > \left[\tau_j - \frac{3}{8}T\right] + < V_j > \left[T - \tau_j\right]$$

if c, contribution as

$$< V_j > [\tau_{j+1}] + < V_{j+1} > \left[\frac{3}{8}T - \tau_j\right]$$

if d, contribution as

$$<V_j>\left[\frac{3}{8}T\right]$$

1.3.5.2 An Algorithm for an X-Array

Processing the τ values (τ_1, τ_2) from an x-array (or other configurations for which the τ cannot be directly related to a velocity magnitude) requires a computing algorithm that includes, but goes beyond, the regularization noted above. One such algorithm (attributed to Scott C. Morris in terms of processing C-CTA signals) is provided here.

Consider that each τ value of an x-array represents a combination of velocity magnitude (Q, in the plane of the x) and the pitch angle with respect to the probe shaft. The latter is designated as γ. Namely

$$\tau_1=\tau_1(Q, \gamma) \tag{1.3.13}$$

and

$$\tau_2=\tau_2(Q, \gamma). \tag{1.3.14}$$

Figure 6 shows the family of $\tau(Q)$ values for the discrete γ values (-36, -30, -24…0…24, 30, 36) of the full calibration data set. The Morris strategy (developed for the equivalent $E^2_{\text{Bridge}}(Q)$ for discrete γ values) involves the equivalent of plotting the 13 distinct Q values that correspond to the measured τ (or E^2) values from the two sensors and identifying their intersection point as shown in Fig. 7. For the conventional anemometer, the sampled voltages are "instantaneous" and the Q and γ values are similarly defined for this very short time segment.

The need to register the two PWM-CTA channels to each other presents a different issue. Namely, the τ_1 and τ_2 samples must be co-registered and then the Q, γ values can be evaluated. For this, the Q value corresponding to γ=0 will be used for the τ_1 and the τ_2 time series. This will then provide

$$\tau_1 \Rightarrow Q_1(\gamma=0) \tag{1.3.15}$$

and

$$\tau_2 \Rightarrow Q_2(\gamma=0) \tag{1.3.16}$$

where these are regularized to a center point of

$$\left[t_j + \frac{7}{8}T\right]$$

The calibration data, as shown in Fig. 6, can be further processed to create a two-equation, two-unknown solution as:

$$Q(\gamma) = Q_1(0)\left[\frac{Q(\gamma)}{Q(0)}\right]_{\tau_1} = Q_1(0)\,f_1[\gamma] \tag{1.3.17}$$

and

$$Q(\gamma) = Q_2(0)\left[\frac{Q(\gamma)}{Q(0)}\right]_{\tau_2} = Q_2(0)\,f_2[\gamma] \tag{1.3.18}$$

for which there will be a unique

$$Q, \gamma$$

combination. One can think of the calculation for (Q,γ) as identifying the intersection point of two curves in the (Q,γ) plane. If sensor 1 is oriented such that $\gamma=+45^o$ is perpendicular to the sensor and if $\gamma=-45^o$ is perpendicular to sensor 2, then

$f_1(\gamma)$ is a monotonically decreasing function
$f_2(\gamma)$ is a monotonically increasing function

and, plotting the curves (1.3.17) and (1.3.18) will lead to a unique intersection point. Note that the smoothed calibration data: $\tau/T=g[Q;\gamma]$, permit arbitrarily fine steps in Q(0) to define the respective "f" functions. For example

$$f[\gamma;Q(0)]=a+b(\gamma)+c(\gamma^2)+\ldots$$

where a=a[Q(0)]. b=b[Q(0)], etc., and the a,b,c…coefficients are obtained from the (for example) 13 γ conditions: 0, $\pm 6^o$, $\pm 12^o \ldots \pm 36^o$.

For reference in the present ***(30 April 2003)*** version of the manual, it can be noted that if

$$Q_1(\gamma=0) \ @\ \tau_1 > Q_2(\gamma=0)\ @\ \tau_2$$

then $\gamma>0$. Conversely, if

$$Q_1(\gamma=0)\ @\ \tau_1 < Q_2(\gamma=0)\ @\ \tau_2$$

then $\gamma<0$.

1.4. Observations Regarding the C-CTA cf the PWM-CTA Operation

1.4.1. The PWM-CTA has a second power

$$\frac{\tau}{<T>} \sim V^n$$

vs. a fourth power (C-CTA)

$$E^2 \sim V^n$$

transfer function. Hence, it is a more sensitive device.

1.4.2 Unlike the C-CTA, the PWM-CTA is inherently stable (electronically).

1.4.3 There is no inherent upper frequency response:

"The T value can be decreased in proportion to the increasing V value". for the PWM-CTA. The C-CTA is inherently limited by its Gain/Bandwidth amplifier characteristic.

Note: for a given master clock frequency, the resolution of the τ values will be limited as T is decreased. However, the analog electronics can be relied upon to function well up to response times of as fast as 0.2 μsec depending on gain and bandwidth settings for the present T=10 μsec unit.

2.0 Sub-units and Inter-connections

2.1 Main Chassis

The main chassis of the unit generates power and cooling for the individual system boards. The front panel of the main chassis is suitable for up to 18 analog channels. When delivered the front panel is configured for the number of channels initially ordered. However, the back side of the panel is marked to allow the addition of channels up to a total 18 channels in the panel. In addition, the front panel supports the clock generator board for the system. Note that the clock generator board also will support up to 18 channels. Again, it is delivered with only the initial clock outputs populated. However, these can be populated by a simple factory upgrade.

In addition to the front panel, the main chassis also houses the system power supply. The system power supply is configured to run with an input of either 110/120 or 220/240 VAC (50 or 60Hz). However, the supply voltage must be set using the jumper card within the power inlet module. Failure to properly set this jumper may result in damage to the unit. Such damage will void the warrantee. See Figure ??? for jumper configuration.

The main chassis also provides cooling for the system. As a result, it is important not to block the cooling passages, both on the top of the chassis and beneath the chassis.

2.2. **Master Board**

The master board provides the PWMCTA with the intelligence necessary to interpret the input control signals generated within the host PC. It also sets the sampling frequency of the unit and controls data acquisition via the control I/O cable as well as generating the data signals to the host PC. It contains a user configurable jumper to set the baud rate (frequency) for the COM port I/O. The board supports standard frequencies of 4800, 9600, 19,200, and 38,400 baud. The unit is factory configured for 38,400 baud. This may be changed by J7 on the master board. The system is hard configured for 1 start bit, 1 stop bit, and 8 data bits. These three parameters cannot be changed.

Jumper 7-0 (rear of card)	Jumper 7-1 (front of card)	Baud Rate
Absent	Absent	4800
Absent	Present	9600
Present	Absent	19,200
Present	Present	38,400

2.3. **PWM Analog Board**

The PWM Analog Board provides the "real" PWMCTA function. This board is used to measure cold resistance, set the amplitude of the pulse waveform and to control the duration of the pulse by comparing the probe voltage to a fixed reference. Alternatively, the PWM Analog Board can allow the user to access its internal 16 bit A/D converter to record voltages such as would be generated by a pressure transducer or temperature sensor.

Each PWM Analog Board in your PWM-CTA must be assigned a specific address via jumpers J4 located on the analog boards. These jumpers are preset at the factory beginning with channel 0 on the far left of the front panel. The next channel to the right is set as channel 1, and so forth. Unless the user is adding more channels, there should be no reason to change these jumpers. The jumper to the rear on each card is weighted as 2^0. As one progresses to the front of the card, each jumper increases in weighting, such that the front most jumper has a weighting of 2^4.

2.4. **Datel Card**

The Datel card provides the high-speed data acquisition function for the PWM-CTA. The card is a PCI-30259 purchased from Datel, Inc. and modified to meet the requirements of DFTI for the PWM-CTA. The modifications include the addition of a plug-in daughter card to translate the proprietary high-speed serial format as sent by the PWM-CTA. Note: the PCI-30259 requires a full-length (12.25" minimum length) PCI slot.

2.5. **Cables**

2.5.1 **Data I/O Cable**

The system communicates via a special I/O cable. This cable plugs into the PWM-CTA on one end. The PC end connects to a special plug that is mounted on an I/O slot cover for the PC. This slot cover may be removed and discarded if the user desires. The special plug contains three connections. The first and simplest connects to the com port on the PC. The remaining two provide serial data and clock connections to the Datel daughter card. See Figure ? for further details.

2.5.1 **Probe Cables**

The probe cables are made especially for the PWM-CTA. They contain two connections. One connection provides a signal to the Probe and the second connection returns the probe voltage to the PWM-CTA. By using this dual probe cable, the effects of probe cable resistance on overheat, etc. is largely eliminated. Note: the probe cable may be used to monitor other resistance based temperature sensors such as platinum RTD devices. See Figure ? for further details on probe cable configuration.

2.5.2 Probe Cable Configuration

Figure 2.5.2.1 Probe Cable Configuration

3. Front Panel Configuration

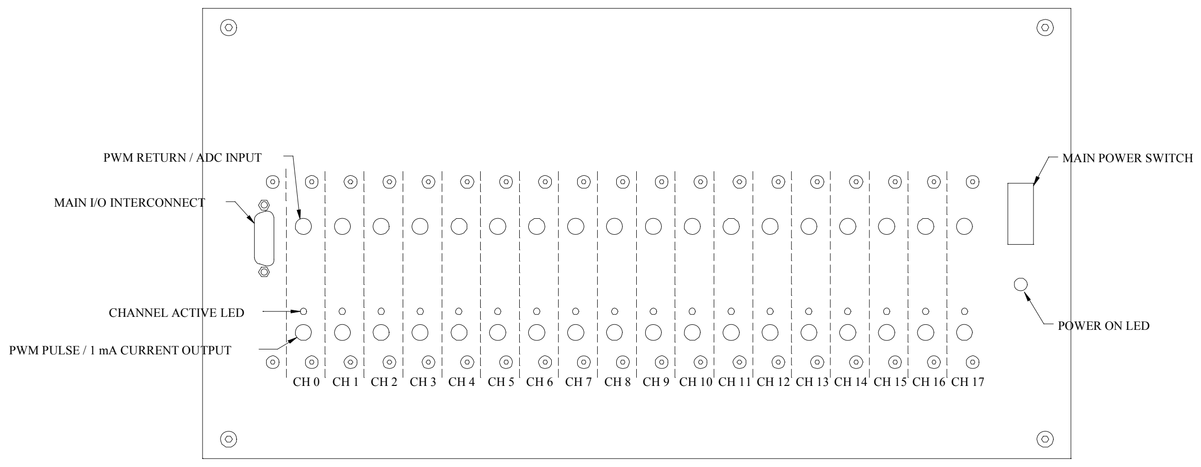


Figure 3.1 Front Panel Configuration

4. Typical Interconnection Diagram

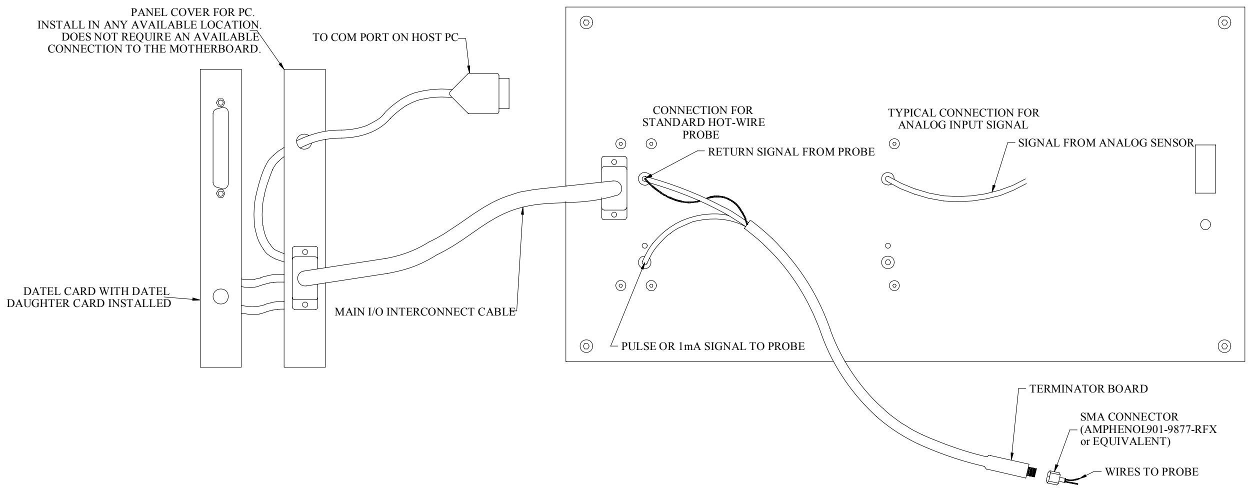

PANEL COVER FOR PC.
INSTALL IN ANY AVAILABLE LOCATION.
DOES NOT REQUIRE AN AVAILABLE
CONNECTION TO THE MOTHERBOARD.
TO COM PORT ON HOST PC
DATEL CARD WITH DATEL DAUGHTER CARD INSTALLED
MAIN I/O INTERCONNECT CABLE
CONNECTION FOR STANDARD HOT-WIRE PROBE
RETURN SIGNAL FROM PROBE
TYPICAL CONNECTION FOR ANALOG INPUT SIGNAL
SIGNAL FROM ANALOG SENSOR
PULSE OR 1mA SIGNAL TO PROBE
TERMINATOR BOARD
SMA CONNECTOR (AMPHENOL901-9877-RFX or EQUIVALENT)
WIRES TO PROBE

Figure 4.1 Typical Interconnect Diagram

6 System Block Diagram

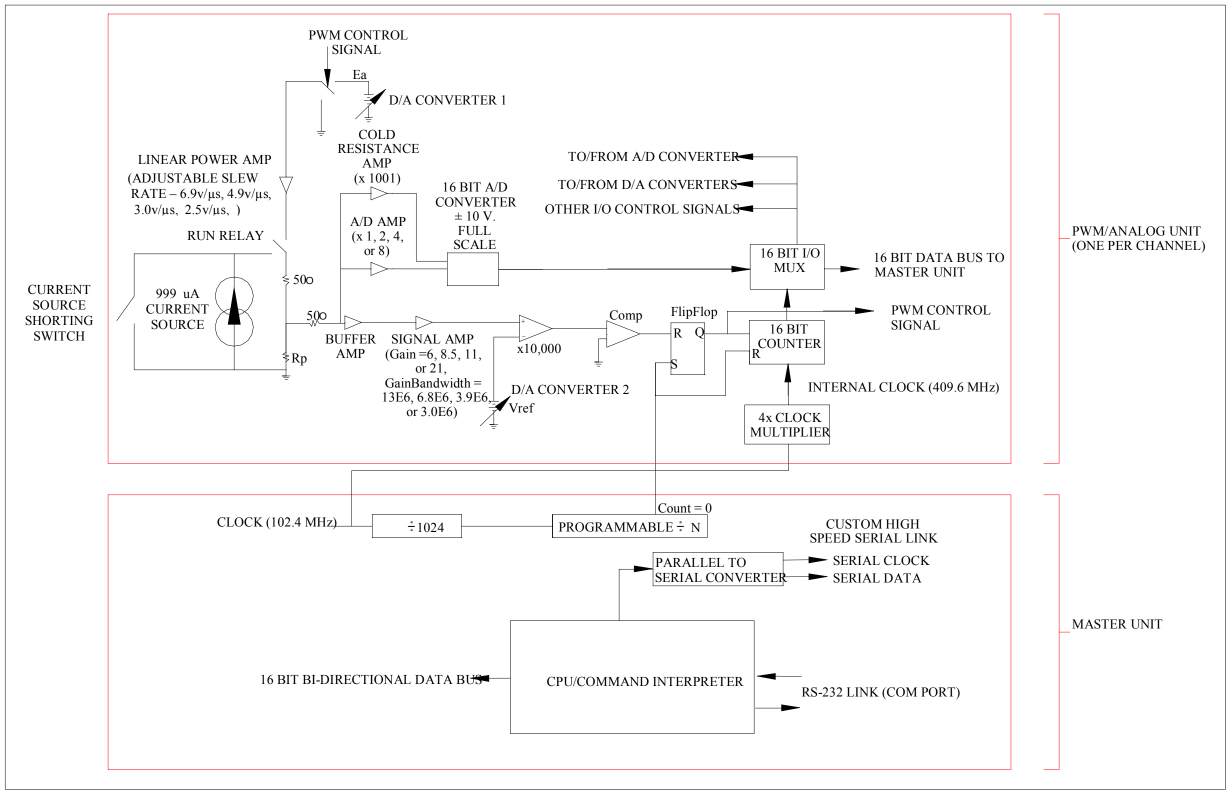

PWM CONTROL SIGNAL
Ea
D/A CONVERTER 1
COLD RESISTANCE AMP (x 1001)
LINEAR POWER AMP (ADJUSTABLE SLEW RATE – 6.9v/µs, 4.9v/µs, 3.0v/µs, 2.5v/µs,)
RUN RELAY
CURRENT SOURCE SHORTING SWITCH
999 uA CURRENT SOURCE
50Ω
50Ω
Rp
A/D AMP (x 1, 2, 4, or 8)
16 BIT A/D CONVERTER ± 10 V. FULL SCALE
BUFFER AMP
SIGNAL AMP (Gain =6, 8.5, 11, or 21, GainBandwidth = 13E6, 6.8E6, 3.9E6, or 3.0E6)
x10,000
D/A CONVERTER 2
Vref
Comp
FlipFlop
R Q
S
TO/FROM A/D CONVERTER
TO/FROM D/A CONVERTERS
OTHER I/O CONTROL SIGNALS
16 BIT I/O MUX
16 BIT DATA BUS TO MASTER UNIT
16 BIT COUNTER
R
PWM CONTROL SIGNAL
INTERNAL CLOCK (409.6 MHz)
4x CLOCK MULTIPLIER
PWM/ANALOG UNIT (ONE PER CHANNEL)
CLOCK (102.4 MHz)
÷1024
PROGRAMMABLE ÷ N
Count = 0
CUSTOM HIGH SPEED SERIAL LINK
PARALLEL TO SERIAL CONVERTER
SERIAL CLOCK
SERIAL DATA
MASTER UNIT
16 BIT BI-DIRECTIONAL DATA BUS
CPU/COMMAND INTERPRETER
RS-232 LINK (COM PORT)

Figure 5.1 System Block Diagram

5. Initial Software Configuration
   5.1. Load the Datel software
   5.2. Load the DFTI software
   5.3. Set the Datel buffer size
   5.4. Check to see that the voltage setting is correct for your country.
   5.5. Connect the PWM-CTA to the main power line
   5.6. Observe that the cooling inputs and outputs on the PWM-CTA are not blocked
   5.7. Turn the main power switch on
   5.8. Observe that the power LED lights and the cooling fans are functioning
   5.9. Connect the I/O cable to the PWM-CTA

6. Operational Sequence

   6.1. Connect the probe cables to any hot-wire channels

6.2. Connect hot-wires to the probe cables as required

6.3. Connect any other sensors to the PWM as required

6.4. Start PWM COMM program.

6.5. Click settings

6.6. Select the com port that will be utilized.

6.7. Click on File/channel setup or press CTRL-N. The screen shown will be used to generate an experiment file to describe the system setup. This is a file with a name with .exp as the file extension.

6.8. The window labeled as [put description here] is intended to contain an optional short piece of text describing the nature of the experiment.

6.9. In the window labeled as frequency select the desired sample frequency.

6.10. For each available channel

- 6.10.1. The block labeled as title can contain a short descriptor of the nature of the channel
- 6.10.2. In the function block select a function
  - 6.10.2.1. Off – this channel is not being used in this experiment
  - 6.10.2.2. Normal PWM – this channel is intended to drive a standard hotwire sensor.
  - 6.10.2.3. Cold resistance – in this configuration, the channel can monitor either a low resistance sensor or a low voltage (less than 10 mV maximum absolute potential) signal.
  - 6.10.2.4. Test mode – this channel will automatically generate a test signal. The test signal will slowly increment its digital output value by one count per sample. This function may serve to check out the communications link for dropped data points.
  - 6.10.2.5. External A/D function – this function will allow the user to monitor an analog based function. Depending on the A/D range setting selection, the input range may be $\pm$1.25V, $\pm$2.5V, $\pm$ 5V or $\pm$10V.
  - 6.10.2.6. Cold Resistance Offset function – this function shorts the 999uA current source to ground and thus allows the user to negate the effect of any

offset voltage between the two probe connectors on the analog board.

6.10.3. Va – this function allows the user to select an initial guess for V_A. The user enters an initial guess for V_A based on experience with the probe type and maximum flow velocity. This value will be refined later in the setup process.

6.10.4. Vref – this setting should not be entered by the user. It is calculated by the PWM COMM program using other user selectable inputs such as gain, R_{COLD}, V_A, etc. When PWM COMM calculates this value it is automatically updated. For information the formula for calculation of $V_{REF\ is}$ as follows:

$$V_{REF} = V_A \cdot \left(\frac{R_{HOT}}{R_{HOT} + R_{SERIES}} \right) \cdot PWM_{GAIN} \tag{6.1}$$

where: $R_{HOT} = (1+OHR)*R_{COLD}$
R_{SERIES} = Channel Impedance (typically 50Ω. See 6.10.8.)
and PWM_{GAIN} is set to 6, 11, 16, or 21 by the user. (See 6.10.5.)

6.10.5 Gain – this setting allows the user to select a gain for the PWM signal amplifier. The gain setting will be a function of the cold resistance of the sensor, the type of sensor, the maximum flow velocity, and any unknown specifics of the sensor. For a typical 5 micron tungsten sensor with a 3-4 ohm cold resistance the normal value for gain will be 11. For other sensors, a lower or higher gain may be desirable. As long as the system can calculate an appropriate V_{REF} (<12.288 volts), the setting is not critical. However, higher gains will often result in lower PWM noise levels. Also, the bandwidth setting (see G below) is dependent on Gain.

6.10.6 Slew Rate – this value is used to set the slew rate of the PWM signal. This sets the degree to which the PWM drive signal is trapezoidal in nature (as opposed to the theoretical infinite slew rate square wave signal.) Generally, the user should select the fastest slew rate.

However, if crosstalk between wires is observed (for example, in the case of certain multiple element probes), a slower slew rate may reduce or eliminate crosstalk.

6.10.7 Bandwidth setting – this setting allows the user to set the gain- bandwidth product of the PWM signal amp. Initially this value should be set to the highest gain-bandwidth setting. Later in the process, the user will be directed to fine tune this setting to reduce PWM noise levels. A reduced Bandwidth setting will reduce system noise and may reduce the potential of crosstalk.

6.10.8 Channel impedance – this setting allows the user to modify the V_{REF} calculation to compensate for modified probe cables. Typically this value is 50 Ω. However, if an extremely precise setting of overheat is desired, the user may chose to modify this value to compensate for cable resistance. The compensation should increase the value of R_{series} by the resistance of the cable in one direction only. For example, in the case of a typical 5 meter probe cable, the total length of this cable is 10 meters. However, only 5 meters of the cable is in series with R_{series}, and thus only half the resistance has an impact the obtained overheat ratio. The value of R_{series} (and thus of R_{cable}) does not impact the measured value of R_{cold}.

6.10.9 Overheat – the user enters the over-heat ratio here. For example, in the case of tungsten, this is on the order of 1.7. (Note that this definition is in contrast to the OHR definition used elsewhere in this text. This value is 1+OHR.)

6.10.10 RCold – this setting can not be entered directly by the user. It is automatically updated when the user clicks on the setup button.

6.10.11 Setup – this button tells the system to measure cold resistance and calculate the appropriate value for V_{REF}. The values for V_{REF} and R_{COLD} are then automatically updated. Note: this function is only allowed for PWM Channels (See 6.10.2.2 above). The button automatically measures the offset and subtracts it from the perceived cold resistance. The value displayed for cold resistance is the actual value. (See 6.10.2.3 and 6.10.2.6 above)

6.10.12 Repeat section 6.10 as necessary for additional channels.

6.11 The user should then click on the Apply changes button. This will request the user to supply an experiment file name and then transmit the selected values to the PWM-CTA. (See 6.12. File menu – below).

6.12 Alternatively, the user may click on Undo Changes. This will reset the values in the experiment configuration sheet to the last saved values. In that case, these values will not be transmitted to the PWM-CTA.

6.13 File menu

- 6.13.1 New – creates a new experiment file loading the values with the default values. (Experiment File names have the extension **.pwm**).
- 6.13.2 Save – saves the current experiment file under the current name.
- 6.13.3 Save as – saves the current experiment file under a specified new name.
- 6.13.4 Open – loads an existing experiment file.
- 6.13.5 Exit – returns to the main menu.
- 6.13.6 Note: the file name here will also set the initial part of the file name used for automatic data acquisition.

6.14 After completing this process, exit to the main menu.

6.15 Set the channel number to the first hot-wire channel in the PWM-CTA.

6.16 Click on start data. The window will change to stop data.

6.17 Click on run active channels.

6.18 The plot will display data from the current channel in a continuous free running mode (oscilloscope mode).

6.19 The amount of data to be acquired per plot and the vertical axis settings for the plot window may be changed via the graph configuration.

- 6.19.1 To do so first click on "**Stop Graph**"
- 6.19.2 Then right click on the plot window
- 6.19.3 Click on settings button that appears
- 6.19.4 The graph settings window will open
- 6.19.5 Enter new values for both axes as desired

6.19.5.1 The standard PWM scale will automatically set the vertical range to 25% to 50% of **T**(where T = maximum pulse width at the sample frequency)

6.20 Optimization of V_A

6.20.1 Option A – experimental flow field easily available

6.20.1.1 Place the probe in the maximum apparent experimental flow field.

6.20.1.2 The mean, standard deviation, and $\pm$ 2 sigma limits are displayed at the bottom of the plot window.

6.20.1.3 Adjust V_A using the Va/Vref adjust buttons so that the peak observed τ values do not exceed 50% of T.

where T= 409.6e6 / sample frequency

6.20.1.4 An increase in V_A will result in a decrease in mean τ and a decrease in V_A will result in an increase in mean τ. In the case of certain multiple element probes, it may be desirable to target a maximum τ value of approximately 40% to 45% of T in order to reduce the potential of crosstalk between elements.

6.20.2 Option B – experimental flow field not easily available

6.20.2.1 Place the probe in the maximum apparent calibration flow field.

6.20.2.2 The mean, standard deviation, and $\pm$ 2 sigma limits are displayed at the bottom of the plot window.

6.20.2.3 Adjust V_A using the Va/Vref adjust buttons so that the mean τ values do not exceed 50% of T. (As an additional safety factor the user may wish to limit the plus 2 sigma value to less than 50% of T.) An increase in V_A will result in a decrease in mean τ and a decrease in V_A

will result in an increase in mean τ. In the case of certain multiple element probes, it may be desirable to target a maximum τ value of approximately 40% to 45% of T in order to reduce the potential of crosstalk between elements.

6.20.3 Now place the probe in a minimum flow field. Observe the mean τ value.

6.20.4 Set the Bandwidth setting to one level slower. Click on the "Set G-BW/Slew" button to apply the changed values. The mean τ value should not increase by more than 1% of T. (Note: a short-term transient change is expected. The final stable value is the value of interest.) If the value increases more than about 1% of the original mean τ, return to the previous bandwidth value. Otherwise continue to decrease bandwidth until the stable τ value increases excessively. Then return to the last higher bandwidth with an unchanged mean τ value. This adjustment will reduce the PWM noise level to a minimum.

6.20.5 For example…

6.20.5.1 The user places the probe in a minimum flow. The observed mean τ is 1043 counts.

6.20.5.2 He decreases the gain-bandwidth from 13.6 MHz to 6.8 MHz

6.20.5.3 The τ increases to 1045. This is a small increase and is acceptable

6.20.5.4 He then reduces the gain-bandwidth to 3.9 MHz

6.20.5.5 The τ increases to 1065 counts. This increase is excessive.

6.20.5.6 The user should return to 6.8 MHz gain-bandwidth for this channel.

6.20.5.7 **WARNING: if the minimum flow used for this setting is not a "zero flow" condition, the user must maintain this minimum flow as long as any hot-wire probe is active.**

Failure to do so may result in a probe calibration shift and/or damage to probes. Clicking stop all channels will deactivate all probes. An active probe is indicated by illumination of the associated LED.

6.20.6 This function sets the Gain Bandwidth product (GBW) for the amplifier in the PWM. The actual amplifier bandwidth is calculated by dividing the GBW by the PWM gain setting (6, 11, 16, or 21). In most cases the user need not be concerned with the exact value of either GBW or the actual bandwidth of the amplifier. The user should only be concerned with the system level behavior as described in 6.20.4.

6.21 Slew rate settings - In most cases a slew rate of 4.9v/µs is appropriate for a PWM channel with a 5 micron wire running at 100KHz. However, if the system appears to suffer from cross talk it may be beneficial to reduce the slew rate. This will change the mean τ values. This change is acceptable. Cross talk can be demonstrated by an increase in the noise level of a channel when a second channel is activated (in comparison to the noise level of the first channel when no other channel is operational). Alternatively, a sudden shift in the mean value of the first channel when the second channel is operational is often indicative of crosstalk. Note that a slower increase in τ is indicative of thermal crosstalk between the two elements. This is inherent to probes with closely spaced elements and is not a function of the PWM. In most cases thermal crosstalk will disappear when the probe is placed within a flow field.

6.22 Click on stop data.

6.23 Repeat for steps 6.15-6.22 for all PWM channels.

6.24 Click on file menu button.

6.25 Set the parameters for your acquisition file(s).

6.26 Return to the main window.

6.27 Enter the desired acquisition time (per file) in the main window.

6.28 Click on run active channels.

6.29 Wait about 30 seconds for the probes to thermally stabilize
6.30 Click on the file – file number headings.
   6.30.1 The file numbering window will open. The user may change the next file name within this window.
   6.30.2 The files names will automatically be created and incremented for the user.
   6.30.3 The file name will begin with the filename where the experiment was saved (not including the .pwm extension)
   6.30.4 The next several characters will be a number where the number of digits and the next value is set in this window.
   6.30.5 The extension will be **.pwd**.
   6.30.6 Enter the values as described above and exit this menu. To reset the next value at any time merely re-enter this window and enter the next file number as desired.
   6.30.7 The next file to which data will be saved is displayed on the top of the main PWMcomm window.
   6.30.8 For example:
      6.30.8.1 The experiment was saved as my_test.pwm.
      6.30.8.2 The data file settings are 3 digits and the next file # is 46
      6.30.8.3 The next data file will be my_test046.pwd
      6.30.8.4 The file after that will be my_test047.pwd
6.31 Click on acquire data.
6.32 The system will acquire data for the desired time frame. The data will then be automatically saved to the first file. The file name will automatically increment. Repeat as desired.
6.33 **THIS IS WHERE WE NEED TO DESCRIBE THE FREE-RUNNING ACQUISTION MODE BEHAVIOR.**
6.34 Click on stop all channels to stop PWM function.
6.35 ***WARNING:*** The unit should not be turned off if any probe is active. To do so may result in damage to the probe. To

deactivate all probes, simply click on “stop all channels” at any time. Channels with active probes are easily detected by the illumination of the LED for that channel.

Appendix A : Control Sequence Scheme

The 100KHz PWM-CTA uses a standard RS232 serial port (COM port) to control its behavior. If the user desires to write their own control program using a PC or other computer to control the PWM, the following may be of use. Each command string is composed of three 8-bit bytes. This appendix will describe this communications protocol. The command set takes the form of

|BYTE1|BYTE2|BYTE3|

where BYTE1 is a word of the form

|X4|X3|X2|X1|X0|A2|A1|A0|

A2-A0 select the function to be performed from the following chart. X4-X0 are reserved for future expansion needs and should be set to all ‘0’s, however these bits are not checked by the current version of the PWM and so a ‘1’ in any of these positions (**X4-X0**) would be interpreted identically to a ‘0’.

			Byte 1 , A2-A0
A2	A1	A0	Action
0	0	0	Communications Reset
0	0	1	Run
0	1	0	Acquire
0	1	1	Setup Channel
1	0	0	Set Frequency
1	0	1	Pre-load Data
1	1	0	Ping Channels
1	1	1	Reserved For Future Use

Return string:

Unless otherwise stated, all valid command strings will echo back the three characters exactly as received. Any invalid command string of three characters will echo back that same string of three characters.

Communications Reset:

If the first character (**Byte1**) in the command string is a null character (0x00) then the system will immediately reset the communications link. The next character is then again a new **Byte1**. Thus Communications Reset is the only function which does not require a three character input string. A single null character as the first character will immediately reset the communications link. Immediately after this communications reset, the PWM will echo back a series of three null characters. Thus if a series of 3 null characters are sent, the system will perform three communications resets and echo back a total of 9 null characters.

Run Function:

If the first character (**Byte1**) in the command string is 0x01 then the action will be to either run all active channels or to stop all active channels. Thus if any channel is assigned as a PWM channel this function will turn the PWM function either on or off. To activate the PWM function on all PWM channels, **Byte2** is 0x01. To deactivate PWM function, **Byte2** is 0x00. **Byte3** is not utilized but should be set to 0x00. If this function is active all the LED on all PWM channels will be illuminated and all PWM channels will be active (pulsing).

Acquire function:

If the first character (**Byte1**) in the command string is 0x02 then the action will be to either enable or disable the acquire function. To activate the acquire function on all active channels, **Byte2** is 0x01. To deactivate the acquire function, **Byte2** is 0x00. **Byte3** is not utilized but should be set to 0x00. The function will echo back exactly the command string that was received. Note: acquire only controls the data being sent to the PC. All active channels will be transmitted whether PWM, test, R_{COLD}, or Analog-to-digital converter mode.

Setup Channel function:

This function is described later in this appendix.

Set Frequency function:

If the first character (**Byte1)** in the command string is 0x04 then the action will be to set the sampling frequency. In this case **Byte2** is of the form

|X2|X1|X0|F4|F3|F2|F1|F0|

where F4-F0 are the frequency divider in binary form. The sampling frequency is defined as 100 KHz/ div, where div is the binary integer **F4-F0**. Setting sampling frequency also defines maximum resolution (**T**). **T** is defined as 4096 * div. For example, if **F4-F0** are 01000 then div is 8, sampling frequency is 12.5 KHz, and **T** is 32768. **X2-X0** are unused and should be set to ‘0’s, however these bits are not checked by the current version of the PWM and so a ‘1’ in these positions (**X2-X0**) would be interpreted identically to a ‘0’.

Pre-load Data function:

If the first character (**Byte1)** in the command string is 0x05 then the action is to pre-load a data value to be sent to the individual channel. The sixteen bit data is contained in **Byte2** and **Byte3**.

Ping Channels function:

If the first character (**Byte1)** in the command string is 0x06 then the action will be to ping channels. By **ping** we mean that we are trying to determine if a particular channel exists in a PWM unit. In the case of a ping action **Byte2** is of the following format

|F0|X1|X0|C4|C3|C2|C1|C0|

If **F0** is a ‘1’ then the function will ping a single channel as chosen by **C4-C0**. If **F0** is a ‘0’ then the function will ping 16 channels at once. In the case where sixteen channels are being pinged at once **C0** will define which set of sixteen channels is being pinged. If **C0** is a ‘0’ then channels 0-15 will be pinged. If **C1** is a ‘1’ then channels 16 to 31 will be pinged. When **F0** is a ‘0’ then **C3-C0** are ignored and should be set to ‘0’s. **X1-X0** are unused and should be set to ‘0’s.

This function returns two 3 character strings. The first string is the normal echo back of the string as received. The second 3 character string is of the form

Byte1	**|CH15|CH14|CH13|CH12|CH11|CH10|CH09|CH08|**
Byte2	**|CH07|CH06|CH05|CH04|CH03|CH02|CH01|CH00|**
Byte3	**|’1’|’1’|’1’|’1’|’1’|’1’|’1’|’S0’|**

If the function is to ping 16 channels, then **CH15-CH0** will return the status of the 16 channels requested. A ‘1’ corresponds to an existing channel and a '0' corresponds to a non-existing channel. In the case of channels 0-15 the channel number (**CH15-CH00)** will correspond directly to the assigned channel address. In the case of the upper bank of channels (Channel 31 down to Channel 16) an offset of 16 must be added to the channel number returned. For example, if CH15 is a '1' and the upper bank was addressed the channel 31 exists. Likewise, if CH15 is a '0' and the lower bank was addressed the channel 15 does not exist. **Byte3** will be returned as 0xFF for a 16 channel ping.

In the case of a single channel ping **Byte3** will indicate the status of that channel. If **S0** is a '1' then the channel exists, else if **S0** is a '0' then it does not exist. **Byte1** and **Byte2** both return 0xFF for a single channel ping.

Future Use:
If the first character (**Byte1)** in the command string is 0x07 or higher then the action is undefined. These codes are reserved for future expansion.

Setup Channel function:
If the first character (**Byte1**) in the command string is 0x03 then the action will be to set-up channel. In this case **Byte2** is of the form

|X2|X1|X0|C4|C3|C2|C1|C0|

where **C4-C0** is the channel to be configured (0-31). **X2-X0** are unused and should be set to '0's, however these bits are not checked by the current version of the PWM and so a '1' in these positions (**X2-X0**) would be interpreted identically to a zero.

.

In the case of Setup Channel function, **Byte3** is of the form

|X4|X3|X2|X1|X0|S2|S1|S0|

S2-S0 select the function to be performed and are defined by the following chart. **X4-X0** are unused and should be set to '0's, however these bits are not checked by the current version of the PWM and so a '1' in these positions (**X4-X0**) would be interpreted identically to a zero.

Set-Up Channel, Byte 2 , S2-S0			
S2	S1	S0	Sub-function
0	0	0	Channel Off
0	0	1	Normal PWM function
0	1	0	Test Mode
0	1	1	Gain/BW/Slew/ADC Gain/R_offset
1	0	0	R_{COLD} mode
1	0	1	Set V_A
1	1	0	Set V_{REF}
1	1	1	External A/D Converter mode

Channel Off sub-function:

If **Byte2** is 0x00 the function is to turn the channel off. This applies whether the channel is in PWM mode, Test mode, R_{COLD} mode, or External A/D Converter mode.

Normal PWM sub-function:

If **Byte2** is 0x01 the channel will be placed in normal PWM mode. Note that the channel will not PWM unless the main mode has previously been set to Run mode (**Byte1**=0x01, **Byte2**= 0x01, **Byte3**=Don't care).

Test Mode sub-function:

If **Byte2** is 0x02 the channel will be placed in test mode. Test mode forces the channel to generate a test output vector signal. This signal generates an output value of 0x0000 for the first sample, 0x0001 for the second sample and so on. Note: the output at the PC will not likely start at 0x0000 as there is no synchronization between the initial start up of test mode and the start of acquisition.

Gain/BW/Slew/ADC Gain/R_offset sub-function:

If **Byte2** is 0x03 the channel will be set up for PWM Gain, PWM Bandwidth, PWM Slew rate, A/D Converter Gain, and R_offset. These data for these controls must first be pre-loaded via the data pre-load function. The format for this data is

Byte2 of pre-load instruction is
|R_OFFSET|X6|SR1|SR0|X5|X4|BW1|BW0|

Byte3 of pre-load instruction is
|X3|X2|ADC1|ADC0|X1|X0|PWM1|PWM0|

If **R_OFFSET** is a '0' then the apparent cold resistance of the probe is measured when the user calls for cold resistance. If **R_OFFSET** is a '1' then the 999 µA current source is shorted to ground and when the user next calls for the cold resistance gain, the value returned is the cold resistance offset for the channel. **R_OFFSET** is intended to allow the user to compensate for small offset voltages between the two probe connections in each analog.

The slew rate of the output amplifier for the PWM signal is set by **SR1** and **SR0**.

SR0	**SR1**	**Slew Rate**
0	**0**	**6.9V/µS**
0	**1**	**4.9V/µS**
1	**0**	**3.0V/µS**
1	**1**	**2.5V/µS**

In most cases the fastest slew rate will be optimal.

The Gain Bandwidth of the input amplifier for the PWM signal is set by **BW1** and **BW0**.

BW0	**BW1**	**Gain-Bandwidth Product (GBW)**
0	**0**	**13.6 MHz**
0	**1**	**6.8 MHz**
1	**0**	**3.9 MHz**
1	**1**	**3.0 MHz**

The highest slew rate should be selected initially. First the user must adjust V_A for optimal behavior at the maximum apparent flow rate. Then the flow rate should be set to a minimum apparent velocity. The user should first notice the tau values returned with the gain-Bandwidth set to 13.6MHz. The user may reduce the Gain-Bandwidth until an excessive increase in Tau is noted (greater than about 0.5% of T.) The correct Gain-bandwidth setting is the slowest gain-bandwidth setting where no substantial change in mean tau is observed.

For example, first the user places the probe in a maximum flow. The sampling frequency is 100KHz. He then adjusts V_A until the maximum Tau value observed is slightly less than 2048 counts (50% of T, T is 4096 counts.) (See frequency setting for a discussion of T.) He then places the probe in a minimum apparent flow. The observed mean Tau value is 1067 counts with a gain bandwidth setting of 13.6 MHz. At 6.8 MHz the mean Tau value is 1070 counts. At 3.9 MHz the mean Tau rises to 1090 counts. The user should then set the gain-bandwidth for this probe in this channel to 6.8 MHz. He should not proceed to 3.0 MHz. To do so may result in a change in the probe's calibration and/or damage to the probe. This adjustment should only be performed prior to calibrating the probe. The advantage of using slower gain-bandwidth settings is that a slower gain-bandwidth setting will result in a lower noise floor, especially at higher frequencies in the spectral plot. In addition, a slower gain-bandwidth may reduce the potential for cross-talk between elements of certain multi-element probes.

The Gain of the input amplifier for the analog to digital converter is set by **ADC1** and **ADC0**.

ADC1	ADC0	Gain	Range
0	**0**	**1**	**-10 volts to +10 volts**
0	**1**	**2**	**-5 volts to +5 volts**
1	**0**	**4**	**-2.5 volts to +2.5 volts**
1	**1**	**8**	**-1.25 volts to +1.25 volts**

The Gain of the input amplifier for PWM section is set by **PWM1** and **PWM0**.

PWM1	**PWM0**	**Gain**
0	**0**	**6**
0	**1**	**11**
1	**0**	**16**
1	**1**	**21**

The PWM gain setting allows the user to optimize the function of the PWM for various wire types. In general, a high resistance wire will require a lower gain setting. With a typical 5 micron tungsten wire (3-4 ohms cold resistance), a gain of 11 seems to be optimal. However, with a higher resistance wire, it may be desirable to use a gain of 6.

Note **X6-X0** are not interpreted in the current version of the PWM. However, for reasons of future expansion, it is recommended to set these to '0's.

R_{COLD} sub-function:

If **Byte2** is 0x04 the channel will acquire the cold resistance of the sensor. If R_OFFSET is set to '0', 1 volt equates to 1 ohm. If R_OFFSET is set to '1', then the return value is the offset for cold resistance measurement. Thus, the actual cold resistance measurement will always be the perceived cold resistance minus the offset cold resistance. Note: measuring R_{COLD} immediately after stopping a hot probe from PWM will result in a high value for cold resistance. The user should generally wait at least 10 to 20 seconds after turning a probe off before using an obtained R_{COLD} value.

R_{COLD} is returned as follows:

First the PWM echoes the R_{COLD} request. Then the PWM returns three bytes. The first two bytes contain the cold resistance. The third byte is 0x00. The value from the first two bytes is interpreted as follows: the two bytes are combined into a single 16 bit value (r_cold_integer). This value is then interpreted as an unsigned integer. The interpretation function is voltage= ((r_cold_integer/2^16)*20)-10;

For example, the return was…

0x92, 0x53, 0x00

The 16 bit hexadecimal integer is 0x9253.

The decimal equivalent of 0x9253 is 37459.

This calculates to 1.43 volts and, for a cold resistance measurement, 1.43 ohms.

Note that there is a one sample pipeline delay in the returned value. Thus the first request returns the value on that channels bus just prior to the request. Because the function had not yet been set to R_{COLD} the value is invalid. The second request returns the value just prior to that the second request. That value is valid as an R_{COLD} measurement as the function is now set to R_{COLD}. To get an accurate value the recommendation is to take 11 samples, discard the first return and average samples 2 through 11. Similarily, one should discard the first R_COLD_OFFSET value.

Set V_A sub-function:

If **Byte2** is 0x05 the channel will set the voltage for V_A. The data for this voltage must first be pre-loaded via the data pre-load function. The format for this data is

Byte2 of pre-load instruction is
|V15|V14|V13|V12|V11|V10|V09|V08|

Byte3 of pre-load instruction is
|V07|V06|V05|V04|V03|V02|V01|V00|

where V15 down to zero is a 16 bit unsigned integer.
The value of this integer is defined as follows.

Int=((voltage/3)/4.096)*(2^16)

For example,
Assume 7.5 volts for V_A is desired.

The integer from the above equation is 40000 which is 0x9C40.

Therefore; **Byte2** is 0x9C and **Byte3** is 0x40;

Note: the maximum value for V_A is 12.287 volts.

Set V_{REF} sub-function:

If **Byte2** is 0x06 the channel will set the voltage for V_{REF}. The data for this voltage must first be pre-loaded via the data pre-load function. The format for this data is

Byte2 of pre-load instruction is
|V15|V14|V13|V12|V11|V10|V09|V08|

Byte3 of pre-load instruction is
|V07|V06|V05|V04|V03|V02|V01|V00|

where V15 down to zero is a 16 bit unsigned integer.
The value of this integer is defined as follows.

Int=((voltage/3)/4.096)*(2^16)

For example,
Assume 7.5 volts for V_{REF} is desired.

The integer from the above equation is 40000 which is 0x9C40.

Therefore; **Byte2** is 0x9C and **Byte3** is 0x40;

Note: the maximum value for V_{REF} is 12.287 volts.

V_{EXT} sub-function:

If **Byte2** is 0x07 the channel will acquire the voltage at the analog input to the channel. The Analog-to-Digital converter range is determined by the A/D gain settings.

V_{EXT} is returned as follows:

First the PWM echoes the V_{EXT} request. Then the PWM returns three bytes. The first two bytes contain the external voltage. The third byte is 0x00. The value from the first two bytes is interpreted as follows: the two bytes are combined into a single 16 bit value. This value is then interpreted as an unsigned integer. The interpretation function is

voltage= [((integer/2^16)*20)-10]/ADC_gain;

For example, the return was…

0x62, 0x53, 0x00 and ADC gain was 4;

The 16 bit hexadecimal integer is 0x6253.

The decimal equivalent of this is 25171.

This calculates to –0.5796 volts.

V_{EXT} has the same pipeline delay issues as R_{COLD} (see above.)

Transmissions to PC:

The normal message for transmission from the PWM-CTA to the PC is an echo of the message that was sent to the PWM-CTA. The exception to this rule is when the message requests data. In this case the initial message will be returned followed by the 16 bit value returned with a final null (0x00) character. Communications Resets are also handled differently. The communications reset occurs as soon as the first byte of a message is a null character. In order to guarantee that a reset takes place it is necessary to send 3 null characters in a row. This is because the PWM host PC cannot be certain where the PWM "thinks" it is in a sent message (0 , 1, or 2 characters already sent). Upon having received a null character the system is immediately ready for a new complete 3 character message. To signify this to the host PC, the PWM immediately echos a set of 3 null characters. Thus, the response to a series of 3 null characters may be a set of 3, 6 or 9 null characters.

Appendix B - Frequency Setting Information

Divider Setting	Sample Frequency (Hz)	Maximum Tau Value (T)
0	100000	4096
1	50000	8192
2	33333	12288
3	25000	16384
4	20000	20480
5	16667	24576
6	14286	28672
7	12500	32768
8	11111	36864
9	10000	40960
10	9091	45056
11	8333	49152
12	7692	53248
13	7143	57344
14	6667	61440
15	6250	65536
16	5882	69632
17	5556	73728
18	5263	77824
19	5000	81920
20	4762	86016
21	4545	90112
22	4348	94208
23	4167	98304
24	4000	102400
25	3846	106496
26	3704	110592
27	3571	114688
28	3448	118784
29	3333	122880
30	3226	126976
31	3125	131072

Appendix C - Data format

The data is transmitted in order of activated channel number. Inactive channels are not transmitted. At transmission, there is no assignment of channel number to any data value. Thus it is imperative to keep track of which channel are active and which channel is transmitted first. To do so the PWM will always activate and deactivate acquisition (Acquire Data and Stop Data buttons) beginning with the first channel and ending with the last channel. The data is transmitted in "big-endian" format. Data is transmitted in 16 bit format. The analog data is transmitted in an unsigned (not twos-complement format). This is true for both data transmitted to the PWM-CTA and data received from the PWM-CTA. The analog conversion functions are:

DtoA_volts = [transmitted_number/(2^16)]*12.288
Transmitted_number=(DtoA_volts/12.288)*(2^16)

AtoD_volts=(([received_number/(2^16)]*20)-10)/AtoD_gain
Received_number=((AtoD_volts*AtoD_gain)+10)*20

For example, channel 0 is PWM. Channel 4 is PWM. No other channels are active. Sample frequency is 50KHz and analog gain on channel 4 is 4.

The following data is received:
0DCE
4EF2
0CAD
3E2A
0ADE
4004
….

First remember that all data is unsigned format (16 bit values are 0 to 2^{16}-1, not -2^{15} to (2^{15}-1)). So convert to unsigned integers…
3534
20210
3245
15914
2782
16388…

The first data word is 3534. This represents a PWM-CTA output from channel 0. For a sample frequency of 50KHz "T" is 8192. The resultant Tau/T is thus 0.431396. This value would then be converted to a velocity based on a modified Collis and Williams calibration. The modification is that the calibration is based on tau/T instead of E^2. Thus $\text{Tau/T}=A+B^*\text{Velocity}^n$.

The second value represents a voltage from channel 4. This voltage converts to –0.9581 volts.

The remainder of the values can be converted similarly. Thus the original data set converts to…

Original value	Converted value	Source
0DCE	0.431396	Channel 0 (tau/T), data point 1
4EF2	-0.9581 (volts)	Channel 4 (volts), analog input voltage point 1
0CAD	0.396118	Channel 0 (tau/T), data point 2
3E2A	-1.28586 (volts)	Channel 4 (volts), analog input voltage point 2
0ADE	0.3396	Channel 0 (tau/T), data point 3
4004	-1.24969 (volts)	Channel 4 (volts), analog input voltage point 3

Appendix d - Matlab Processing Files

File opening – pwm2mat

The file pwm2mat.m is a simple binary to matlab conversion routine. Each column contains the data from a single channel. The file name not including the extension (.pwd) is the first input parameter. Inactive channels are not recorded and should not be included in the numchannels parameter. The return is an array with "numchannels" columns and as many points per column as the file supports.

Calibration - spincal.m

Spincal.m serves to extract calibration parameters from a data set given an input data set extracted using the pwm2mat routine. We calibrate using a transient calibration system where the calibration sweep takes about 50 seconds. The routine takes a set of local means and then uses them to generate A, B, and n. It returns the best A, B, and n and an RMS error as a figure of merit for the calibration. ***WHAT ABOUT THE EFFECT OF SPIN DIRECTION ON THE CALIBRATION?***

Processing – tau2q.m

Tau2q is used to generate a velocity time series from a tau time series. It converts each adjacent tau pair to an irregular velocity time series. It then takes each pair of irregular velocity values and regularizes them in time.

Xwire calibration – spincalx.m

Spincalx.m works in much the same manner as spincal, with the exception that it uses a pair of hot wires (as in an X-Wire pair.) The routine must be called once for each calibration angle.

Reregularization – taureg.m

The taureg.m routine is used to regularize the data from each half of an x-wire before converting to velocity and angle values. This is necessary because the constant energy points for the two wires are not precisely aligned in time. The routine converts the tau values to an irregular apparent velocity series based on the zero degree calibration coefficients. These irregular velocities are re-regularized as in the tau2q routine above. The re-regularized velocities are then converted back to a series of duty-cycles (tau/T). The duty cycles can be processed as if they were regular in time and as if T were one (as opposed to 4096, for example.)

x-wire processing – pwmxprox.m

The pwmxprox.m routine takes the regularized duty cycle values from taureg and converts them to a series of velocities and angles. It also returns a figure of merit in the form of "bad_ratio". Bad_ratio is the fraction of points which cannot be converted to a velocity and angle pair. Some possible reasons for these "failures" could be a velocity which is either excessively high or low or an angle outside the $\pm$36 degree range. The code looks at several alternative methods of curve fitting to get the best values for angle and speed.