

WAVECREST CORPORATION

AG-100TM

USER'S GUIDE AND REFERENCE MANUAL

WAVECREST CORPORATION continually engages in research related to product improvement. New material, production methods, and design refinements are introduced into existing products without notice as a routine expression of that philosophy. For this reason, any current *WAVECREST* product may differ in some respect from its published description but will always equal or exceed the original design specifications unless otherwise stated.

Copyright 2000-2001

WAVECREST CORPORATION
A TECHNOLOGIES COMPANY
7626 GOLDEN TRIANGLE DRIVE
EDEN PRAIRIE, MN 55344
(952) 831-0030
(800) 733-7128
www.wavecrest.com

All Rights Reserved

Patents Pending

Windows is a registered trademark of Microsoft Corporation. Other company names and product names are trademarks or registered trademarks of those companies.

This page intentionally left blank.

AG-100™ User's Guide and Reference Manual

Table of Contents

Preface	Purpose of this Manual.....	vi
	Organization	vi
SECTION 1 Introduction		
	Features	1
	AG-100 Specifications	2
	Environmental Specifications.....	2
	Cleaning Instructions.....	2
SECTION 2 Description		
	Trigger Mode.....	3
	Speed Switch	3
	Protocol Select Switch	3
	Operation Mode Switch.....	3
	Pushwheel Switch	4
	Balun	4
	Amplifiers.....	4
	Pickoff and Associated Ports.....	4
	Arm Out.....	4
	Arm Active.....	4
	Comma Detect.....	4
SECTION 3 GPIB Commands		
	IEEE.2 Common Commands	5
	AG-100 Specific Commands.....	5
	CYCLES.....	5
	COMMA DETECTION	6
	FINE DELAY	6
	HEADER.....	6
	LONGFORM.....	7
	MASK	7
	PATTERN.....	7
	ROCKER SWITCHES.....	7
	SHIFTER BYPASS.....	8
	VERSION.....	8
	VOLTAGE REFERENCE	8
Appendix - Application Programming Interface (API) Commands.....		9

This page intentionally left blank.

Purpose of this Manual

This manual has been prepared to help the user achieve maximum results from the AG-100™ in the shortest possible time. The manual's physical size, format and content is designed to aid the new user as well as the established user.

Organization

The manual is divided into three sections. The following is a brief description of the information contained in each section. For quick reference to specific information, refer to the Table of Contents.

SECTION 1 — *Introduction*

This section introduces you to the AG-100. Capabilities and features are described. A complete table of the performance specifications is included in this section.

SECTION 2 — *Description*

This section describes the functional features of the AG-100.

SECTION 3 — *GPIB Commands*

Includes various parameters, values and codes necessary for implementing remote control of the AG-100 via the IEEE-488 Interface.

APPENDIX - *Application Programming Interface (API) Commands*

Includes structure and function commands necessary for direct access to *WAVECREST's Virtual Instruments Signal Integrity™ 6.02* software. Please refer to the API User's Guide (part no. 200002-02 REV B) for a detailed explanation of the API software.

This page intentionally left blank.

SECTION 1 - Introduction

The **WAVECREST AG-100™** produces a pattern marker from a repeating, markerless data pattern that can then be used as an external arming signal for a **WAVECREST** DTS system. When using a pattern marker as an external Arm signal, **WAVECREST** DTS systems are able to make accurate jitter measurements on a given data pattern using the dataCOM tool in **WAVECREST's Virtual Instrument Signal Integrity™ 6.02 (VISI)** software. Based on pattern match, this Arming signal can be generated for Fibre Channel, Infiniband™ and Gigabit Ethernet signals. For Fibre Channel and Gigabit Ethernet protocols, the **AG-100** works at 1X or 2X speeds. For Infiniband (2.5 Gb/s), the unit must be set to 2X Gigabit Ethernet. For further jitter information, refer to the current *Fibre Channel Methodologies for Jitter Specification* document at www.t11.org.

FEATURES

- Pattern Marker** - Provides pattern recognition of the data Start of Frame (SOF) or any unique 40-bit sequence and then outputs a marker to be used as an arming signal for the DTS-207x products.
- Edge Count Mode** - Edge count mode allows a marker to be generated after a user specified number of edges.
- Built-in amplifiers** - Used when the input signal is below the required input voltage. Two amplifiers are included, each with a nominal gain of 12dB.



Figure 1. AG-100 shown with DTS-2077

AG-100™ SPECIFICATIONS

Data Rate Range for Pattern Match	1.0625, 1.25, 2.125, 2.5 Gb/s
Pattern Match Length Edge Count Mode	up to bits
Programmable Delay Range	up to 39 cycles @1X speeds up to 78 cycles @ 2X speeds
Programmable Delay Resolution	30 ps
Maximum Data Edge Count	2 ³⁴ -1
Maximum Input Voltage at the Pick-Off	3.8 Volts peak-to-peak
Minimum Input Voltage at the Pick-Off	500 mV (peak-to-peak)

Pick-Off Ratio	10 dB Nominal
Pick-Off Insertion Loss (seen at Ch. Out)	3.3 dB Nominal
Amplifier Gain	12dB Nominal
Input Impedance	50 Ω
Balun Transmission Loss	Less than 6 dB typical, 30 Mhz to 3 Ghz

ENVIRONMENTAL SPECIFICATIONS

- **Power Requirements**

Voltage.....	100-120/200-240VAC
Frequency	47-63 Hz
Power	Less than 100 W
Mains Supply Voltage Fluctuations.....	Not to exceed ±10% of nominal line voltages.

- **Environmental Requirements - Indoor Use Only**

Elevation (Altitude)	to 6500 ft.
Operating Temperature	Ambient room temperatures of 15°C to 35°C
Humidity	1% to 85% R.H. (Non-Condensing)
Over-voltage Installation Category*	CAT II
Pollution Degree	2 per IEC664

* This equipment should be connected to power supply mains using the power cord provided. Position the instrument so that the rear power switch and cord are easily accessible. Transient over-voltages are to be less than limits defined by installation category II, Annex J of EN 61010-1.

VENTILATION

To provide adequate ventilation for the AG-100™, a minimum of two inches of unobstructed airflow must be maintained on each side.. To maintain the highest accuracy, the AG-100 air intake should not be near sources that would cause ambient temperature fluctuations. Temperature fluctuations greater than 5°C would require the AG-100 to be re-calibrated.

MAINTAINING PROPER INSTRUMENT GROUNDING

Both 120VAC and 230VAC power sources require an earth ground at the power connection. Failure to provide this safety ground could result in improper measurements and/or electrical shock danger to the operator.

Only qualified service personnel should perform maintenance procedures on the AG-100. When maintenance is performed, NO protective earth ground connections should be removed.

NOTE: Where this product is installed at areas requiring main power connections other than the three-pronged North American power plug, the main power cord must be replaced with the cord and plug appropriate to the location. The main supply cord must be rated to meet or exceed the current rating on the IEC connector (10 amps) and must meet the requirements of H05VV power cords in IEC 227 or IEC 245. If the appropriate power cord cannot be obtained locally, contact the *WAVECREST CORPORATION* factory and one will be provided.

FUSE REPLACEMENT

The IEC plug-in at the back of the AG-100 provides the input connection for the AC power cord. A small compartment on the IEC plug houses two Wickman 4AT/250V fuses (5x20mm). To gain access to the fuses, remove the power to the AG-100 and disconnect the power cord. With a small screwdriver, pry open the fuse compartment on the IEC plug. Remove the fuses and install the new fuses prior to closing the compartment and reinstalling the power cord.

CLEANING INSTRUCTIONS

Periodic cleaning may be required. Use a clean, lint free cloth to dust outer surfaces after electrical power has been removed. If needed, dampen cloth with light amounts of water or alcohol.

INTENDED USE OF EQUIPMENT

The AG-100 should be used only for its intended purpose as outlined in this manual. To avoid possible injury, the AG-100 should not be operated with the top cover or other panels removed. Refer installation and maintenance to qualified service personnel.

To avoid explosion, do not operate the AG-100 in or near an atmosphere of explosive gases

This section describes the functional features of the AG-100™.

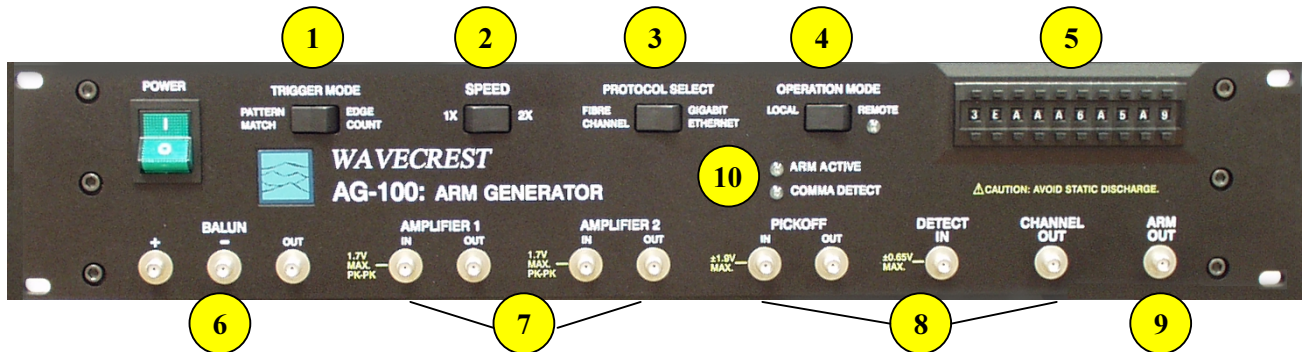


Figure 2. AG-100 Front panel

- 1 Trigger Mode - Pattern Match/Edge Count Switch**
 Selects between Pattern Match and Edge Count modes.
 The AG-100 operates in one of two modes: Pattern Match or Edge Count. Each mode produces an arming signal at the ARM OUT output, however, each mode uses the internal circuitry of the AG-100 in a different way. Pattern Match mode is used when the pattern has a unique, repeating 40-bit word in the pattern. In the case of Fibre Channel compliant patterns the Start Of Frame (SOF) header can be used. For example, for CRPAT, the SOF is "3EAAA6A5A9". To have the AG-100 generate a marker using CRPAT, the pushwheel switches would be set to "3EAAA6A5A9" or input from the GUI interface in *Virtual Instrument Signal Integrity™ 6.02* would be used. Every time the AG-100 recognizes this unique 40-bit pattern, it will produce an arm signal. Pattern marker mode can be used manually or with *VISI 6.02* software.
 Other commonly used patterns are shorter and have no SOF. In these cases, the patterns themselves are entered into the pushwheel switches. For example, an idle pattern is only 40 bits long. Those 40 bits repeat only once each pattern, so in this case, the actual pattern is entered into the pushwheel switches. For an idle pattern, the pushwheel switches would be set to "3EAA2AAAAA" or a shifted version of this. For K28.5, which is 20 bits long, the pattern is entered twice. The pushwheel settings for K28.5 would be "C14FAC14FA."
 Edge Count mode is used when pattern lengths are not integer multiples of 20. For example, Edge count mode generates an arm signal using a pseudo-random bit stream (PRBS). The user can input the number of positive or negative edges in the pattern from the front panel or from the *VISI 6.02* GUI screen. From the *VISI 6.02* screen, the user can also input the pattern and the software will automatically determine the number of positive or negative edges in the pattern.
- 2 Speed Switch**
 Selects between 1X and 2X speeds for Fibre Channel or Gigabit Ethernet. Switch must be set to 2X for Infiniband™.
- 3 Protocol Select Switch**
 Selects either Fibre Channel or Gigabit Ethernet. For Infiniband the switch must be set to Gigabit Ethernet.
- 4 Operation Mode Switch**
 Selects Remote (*VISI 6.02*) or Local (Front Panel) operation. Remote operation allows control of the AG-100 using a GPIB command set, such as, *VISI 6.02* software.

5

Pushwheel Switch

In Pattern Match mode, the switch settings are set to match a unique 40-bit sequence in the pattern. In a typical compliant pattern, this would be the Start Of Frame (SOF) header. In Edge Count mode, only 8 characters are used, starting from the right. These characters are used to set the divider ratio. In general, Edge Count is only used in conjunction with the *VISI™ 6.02* software. When *VISI 6.02* is used, the manual pushwheel switch is disabled.

6

Balun

The internal balun can be used to convert differential signals to single-ended signals. It can also be used to resolve signal inversion issues with an inverting amplifier by switching the input (+) and input (-) ports on the balun. The result is that the differential data signal gets inverted twice (once by the balun and once by the amplifier). The net result is a non-inverted data stream.

7

Amplifiers

Two amplifiers are available for signal amplification. The amplifiers can be used to boost the input signal before the pickoff (in which case the amplifier will be in the measured signal path) or to boost the signal to the Detect In port after the pickoff (in which case the amplifier will not be in the measured signal path). If the signal at the PICKOFF IN port is less than 250 mV_{p-p}, it is recommended that an amplifier be used. The amplifiers are 50Ω RF amplifiers with a gain of ~12dB. This corresponds to a voltage gain of about 4X. The amplifiers have a 3GHz, 3dB. **It is very important to note that the amplifiers will invert the signal in the time domain.** If an amplifier is used before the Detect In port, the pattern on the pushwheel switches needs to be inverted. For example, if the pattern match was "3EAAA6A5A9" for a non-inverted pattern, the pattern match for the inverted pattern would be "C155595A56." This pattern would be entered on the pushwheel switches or changed in the software to produce an arm signal from an inverted pattern. Another way to correct a data pattern inversion is to pre-invert the data signal using the balun on the front panel. This is done by inserting the positive half of the data signal into the negative port of the balun and the negative half of the data signal into the positive port. If a differential probe is being used, the signal can be pre-inverted by flipping the probe polarity.

8

Pickoff and associated ports

A high-bandwidth pickoff is used to couple the input data signal into the **AG-100™** to produce an arming pulse. The PICKOFF IN port is the input data port. The CHANNEL OUT port of the pickoff is the data output port, which is connected to the DTS measurement channel. The insertion loss of this path is 1 dB.

For proper operation, the CHANNEL OUT port needs to be terminated at 50Ω when it is not connected to a DTS system. The PICKOFF OUT port is the coupled, or pickoff, port. The insertion loss of this path is 10 dB. The PICKOFF OUT port connects to the DETECT IN port. If the signal level of the PICKOFF OUT port is too low (due to the 10dB loss of the pickoff path), the signal should be amplified using one of the amplifiers prior to connecting it to the DETECT IN port. If an amplifier is used, the signal inversion needs to be accounted for.

9

Arm Out

The arm signal that is generated by the AG-100 comes out of the ARM OUT port. The ARM OUT port is connected to one of the external arm ports of the DTS.

10

Arm Active

The ARM ACTIVE light will come on when the AG-100 recognizes a pattern and is outputting an arm signal.

Comma Detect

The Comma Detect light will come on if Comma Detect is enabled and the pattern contains comma characters. Comma sequences are used in the Fibre Channel protocol for synchronization purposes. Comma detect should only be enabled for patterns over eighty bits long.

SECTION 3 - GPIB Commands

The following GPIB commands are supported in the AG-100™.

IEEE.2 Common Commands

- *CLS - Clear Status
- *ESR? - Event Status Register Query
- *IDN? - Identification Query
- *OPC - Operation Complete Command
- *OPC? - Operation Complete Query
- *RST - Reset
- *STB? - Status Byte Query

AG-100 SPECIFIC COMMANDS

• CYCLES

The **CYCLES** command sets the "cycle-based" delay of the AG-100's ARM OUT output signal. This command is useful **ONLY** when operating in Pattern Match mode. Although the command syntax requires that 4 HEX digits be sent, only the lower fourteen bits are significant. Restrictions on the legal values are as follows: **one and only one of the four bits in the subslice (13:10) must be high at any given time, and one and only one of the ten bits in the subslice (9:0) must be high at any given time.**

Command Syntax- :CYCDly<4 character ASCII HEX value>

- Example 1: Send (0, 7, ":CYCD0401", 9, Nlend); //sets cycle-based delay to 0 ps (power-up default)
- Example 2: Send (0, 7, ":CYCD2200", 9, Nlend); //sets cycle-based delay to 36706 ps (FC)
//or 31200 ps (GBEN)
- Example 3: Send (0, 7, ":CYCD1008", 9, Nlend); //sets cycle-based delay to 13176 ps (FC)
//or 11200 ps (GBEN)

Query Syntax - :CYCDly?

- Example: Send (0, 7, ":CYCD?", 6, Nlend);

The following tables show the time delay versus subslice value (Pattern Match mode).

Subslice (13:10) value	# of 1.0625 GHz (FC) cycles OR # of 1.25 GHz (GBEN) cycles	Resulting Delay – FC (ps)	Resulting Delay – GBEN (ps)
0001	0	0	0
0010	1	941	800
0100	2	1882	1600
1000	3	2824	2400

Subslice (9:0) value	# of 1.0625 GHz cycles OR # of 1.25 GHz cycles	Resulting Delay – FC (ps)	Resulting Delay – GBEN (ps)
0000000001	0	0	0
0000000010	4	3765	3200
0000000100	8	7529	6400
0000001000	12	11294	9600
0000010000	16	15059	12800
0000100000	20	18824	16000
0001000000	24	22588	19200
0010000000	28	26353	22400
0100000000	32	30118	25600
1000000000	36	33882	28800

The delay obtained depends upon the protocol selected(FC or GBEN). The delay obtained does NOT depend upon the speed (1X or 2X). The delays obtained from each subslice (13:10 and 9:0) are additive. The "cycle-based" delay circuitry is disabled in Edge Count mode; when in Edge Count mode, this command has no effect.

• COMMA DETECTION

The **COMMA DETECTION** command enables/disables "comma detection" in the front end electronics of the **AG-100™**. When enabled, the successful capture of "comma" characters in the data stream is indicated by the AG-100's COMMA DETECT front panel LED. **COMMA DETECTION must be set to '0' (disabled) whenever the data pattern is less than 160 bits long.**

Command Syntax - `:CDET<ON|OFF|1|0>`

Example: `Send (0, 7, ":CDET1", 6, NlEnd); //turns comma detection ON`

Query Syntax - `:CDET?`

• FINE DELAY

The **FINE DELAY** command sets the "analog" delay (also called the "vernier" delay) of the AG-100's ARM OUT output signal. The delay imposed by this setting is in addition to any "cycle-based" delay. Total analog delay range is 4 nanoseconds.

Command Syntax - `:FDLY<2 character ASCII HEX value>`

Example 1: `Send (0, 7, ":FDLY00", 7, NlEnd); //sets analog delay to 0 ps (power-up default)`

Example 2: `Send (0, 7, ":FDLYFF", 7, NlEnd); //sets vernier delay to 4000 ps`

Example 3: `Send (0, 7, ":FDLYab", 7, NlEnd); //sets vernier delay to 2682 ps`

The weighting of each bin is identical and is equal to approx. 15.686 picoseconds. There is no query available for this command.

• HEADER

The **HEADER** command allows the option of not having the header returned on a response from the AG-100.

Command Syntax - `:SYSTem:HEADer<ON|OFF|1|0>`

Example: `Send(0,5,":SYSTem:HEADerOFF",17,EOI);`

Query Syntax - `:SYSTem:HEADer ?`

Example: `Send (0, 5, ":SYSTem:HEADer?", 15, EOI);`

Response: `<"0"|"1">` (OFF or ON)

- **LONGFORM**

The **LONGFORM** command selects whether the header that is returned from the AG-100™ is of a long form or a short form.

The **LONGFORM** query returns the presently selected long or short form.

Command Syntax- :SYSTem:LONGform<ON|OFF|1|0>

Example: Send (0, 5, ":SYSTem:LONGformOFF", 19, EOI);

Query Syntax- :SYSTem:LONGform?

Example: Send (0, 5, ":SYSTem:LONGform?", 17, EOI);

- **MASK**

The **MASK** command assigns sets a mask value (40-bits long) to determine which bits in the raw match value are significant. A '0' assigned to any mask bit causes the corresponding bit in the raw match value to be treated as a "do care". A '1' assigned to any mask bit causes the corresponding bit in the raw match value to be treated as a "don't care".

Command Syntax- :MASK<10 character ASCII HEX value>

Example 1: Send (0, 7, ":MASK0000000000", 15, Nlend);
//all bits are "do cares" (power-up default), each of 40 successive incoming bits must
//match each successive bit in the raw match value in order to produce an ARM OUT

Example 2: Send (0, 7, ":MASK1111111111", 15, Nlend);
//all bits are "don't cares", not very useful, produces an output every 80 bits

Example 3: Send (0, 7, ":MASK695a820041", 15, Nlend);
// bits 0, 6, 17, 23, 25, 27, 28, 30, 32, 35, 37 and 38 are "don't cares",
// the rest are "do cares"

Query Syntax- :MASK ?

- **PATTERN**

The **PATTERN** command sets the raw match value (40-bits long) to be tested against the incoming data stream.

Command Syntax - :PATTern<10 character ASCII HEX value>

Example 1: Send (0, 7, ":PATT3eaaa6a5a9", 15, Nlend);
//sets a match value useful form compliant patterns like CRPAT and CJTPAT

Example 2: Send (0, 7, ":PATTc14fac14fa", 15, Nlend);
//sets a match value that can be used on the K28.5 pattern

Example 3: Send (0, 7, ":PATT3eaa2aaaa", 15, Nlend);
//sets a match value that can be used with the IDLE pattern

Query Syntax - :PATTern?

- **ROCKER SWITCHES**

The **ROCKER SWITCHES** command allows the user to make three basic operational settings corresponding to the settings invoked (when operating in local mode) by three of the AG-100's front panel rocker switches. Each of the three bits is readable/writeable when operating in remote mode.

TRIGGER MODE SWITCH (Bit 0) – Pattern Match Mode ('0') or Edge Count Mode ('1')

SPEED SWITCH (Bit 1) – 1X ('0') or 2X ('1') speed

PROTOCOL SELECT SWITCH (Bit 2) – Fibre Channel ('0') or Gigabit Ethernet ('1')

OPERATION MODE SWITCH (Bit 3) – not writeable, will always read '1'

Command Syntax - :ROCKers<1 character ASCII HEX value>

Example 1: Send (0, 7, ":ROCKE", 6, Nlend); //sets Pattern Match mode, 2X, Gig Ethernet

Example 2: Send (0, 7, ":ROCK6", 6, Nlend); //also sets Pattern Match mode, 2X, Gig Ethernet

Example 3: Send (0, 7, ":ROCK0", 6, Nlend); //sets Pattern Match mode, 1X, Fibre Channel

Example 4: Send (0, 7, ":ROCK1", 6, Nlend); //sets Edge Count mode

Example 5: Send (0, 7, ":rockf", 6, Nlend); //also sets Edge Count mode

Query Syntax- :ROCKers?

- **SHIFTER BYPASS**

The **SHIFTER BYPASS** command allows/disallows bypass of the AG-100's cycle-based shifter.

Command Syntax- `:SFTByp<ON|OFF|1|0>`

Example 1: `Send (0, 7, ":SFTBOFF", 8, NlEnd); //operation will make use of the shifter`

Example 2: `Send (0, 7, ":SFTBON", 7, NlEnd); //shifter will by bypassed; i.e. no cycle-based delay`

Query Syntax- `:SFTByp?`

- **VERSION**

The **VERSION** query returns the time and compilation date of the AG-100's firmware.

Query Syntax- `:SYSTEM:VERSION ?`

- **VOLTAGE REFERENCE**

The **VOLTAGE REFERENCE** command sets the threshold voltage of the AG-100 input comparators. Actual voltage range is approximately 1.11 volts.

Command Syntax- `:VREF <2 character ASCII HEX value>`

Example 1: `Send (0, 7, ":VREF00", 7, NlEnd); //sets voltage reference to (-)0.555 volts`

Example 2: `Send (0, 7, ":VREFFF", 7, NlEnd); //sets voltage reference to (+)0.555 volts`

Example 3: `Send (0, 7, ":VREF80", 7, NlEnd); //sets voltage reference to (+)0.002 volts
(power-up default)`

Example 4: `Send (0, 7, ":VREF70", 7, NlEnd); //sets voltage reference to (-)0.276 volts`

Query Syntax- `:VREF ?`

APPENDIX - *Application Programming Interface (API) Commands*

Please refer to the API User's Guide (part no. 200002-02 Rev B) for a detailed explanation of the API software.

Structure used for Arm Generator Parameters

```
typedef struct
{
    PARM tParm; // Defaults as follows:
                // Same as PARM in 3.2.4
                // except: lSampCnt = 50
                // and lAutoArm = ARM_EXTRN

    char bPtnBits[ 10 ]; // All zeros
    char bMskBits[ 10 ]; // All zeros
    char sPtnName[ 128 ]; // "sof.ptn"
    long lInvtPtn; // 0
    long lCyclDly; // 0
    long lFineDly; // 0
    long lFunctSw; // 0
    long lSpeedSw; // 0
    long lProtoSw; // 0
    long lCommDet; // 0
    long lCDlyByp; // 0
    long lEdgeCnt; // 0x0F
} AGEN;
```

tParm

This structure contains the DT207x settings to be used when optimizing the marker position with the **ArmFindDly()** function. This is mainly used to specify the Arm and Channel, but may also be used to override default voltage thresholds or other parameters.

bPtnBits, bMskBits

These fields are used to hold the internal representation of the pattern, the details of these fields is not important from an application standpoint. The first time **ArmSetParm()** is called the pattern is loaded into these fields from the file named in the **sPtnName** field. This internal representation is used for all subsequent operations.

sPtnName

Name of pattern file to be used, the file must exist or an error will be returned. The first time **ArmSetParm()** is called, appropriate values are loaded into the **bPtnBits** and **bMskBits** fields. If the pattern file is to be changed, both these fields should be cleared to all zeros so that the new pattern will be loaded on the next call to **ArmSetParm()**. The default file is sof.ptn.

lInvtPtn

Invert the pattern bits if non-zero, this is used to compensate for sending a polarity sensitive signal through an amplifier stage which inverts the signal. This parameter is not used if **lFunctSw** is set to Edge Count Mode.

lCyclDly

Cycle Delay Increment [0 - 39]. The value of each increment is dependent on the protocol. For 1X or 2X Fibre Channel each increment is equal to 941ps. For 1X or 2X GigaBit Ethernet each increment is equal to 800ps.

lFineDly

Fine Delay Increment [0 - 255]. Each increment is equal to approximately 15.686ps, giving a total possible delay of 4ns.

lFunctSw

Marker Generation Function, the following are valid values: 0 Pattern Match Mode 1 Edge Count Mode.

ISpeedSw

Speed Switch, the following are valid values:

- 0 1X Fibre Channel or GigaBit Ethernet
- 1 2X Fibre Channel or GigaBit Ethernet

This parameter is not used if **IFunctSw** is set to Edge Count Mode.

IProtoSw

Protocol Switch, the following are valid values:

- 0 Fibre Channel
- 1 GigaBit Ethernet

This parameter is not used if **IFunctSw** is set to Edge Count Mode.

ICommDet

Enable comma detect in the AG-100's front end SERDES if non-zero.

ICDlyByp

Bypass cycle based delay circuitry if non-zero.

IEdgeCnt

Edge count to be used if **IFunctSw** is set to Edge Count Mode. It should be entered as either the count of positive edges or the count of negative edges (they must be the same), but not the sum of both.

AG-100™ ARM GENERATOR FUNCTIONS

These functions provide access to the *WAVECREST* AG-100™ Arm Generator.

Initialize Arm Generator Device

```
long ArmInitDev ( char *sDevName, long lBrdNumb,  
long lBrdAddr );
```

Input:

- sDevName** Pointer to device name if UNIX platform
- lBrdNumb** GPIB board number
- lBrdAddr** GPIB board address

Return:

- 0 on Success or Error Code on Failure

Example:

```
ArmInitDev ( "dev7", 0, 7 );
```

This function must be called once at the beginning of your application in order to pass information concerning the GPIB configuration. The first parameter is used to specify the GPIB device name on UNIX platforms and is ignored on Microsoft® Windows™. The second parameter is the board number, and the final parameter is the device number. A successful call to `ArmInitDev()` must be accomplished before any other calls to a Arm Generator using the *WAVECREST* Production API.

Typical examples of `sDevName` parameter on SUN Platforms:

<u>sDevName</u>	<u>Description</u>
dev5	Device at bus address 5

Typical examples of sDevName parameter on HP-UX Platforms:

sDevName	Description
hpib, 5	Device at bus address 5, and symbolic name hpib.
7, 5.....	Device at bus address 5, and connected to an interface card at logical unit 7.
lan [128.10.0.3] :hpib, 5	Connect to a LAN server at IP address 128.10.0.3 which contains an hpib interface with device at bus address 5.
lan [hpibsrv.wave.com] :7, 5	Connect to a LAN server named hpibsrv.wave.com which contains an interface card at logical unit 7 with primary device at bus address 5.

Cleanup Prior to Application Termination

```
long ArmExitDev ( void );
```

Input:

None

Return:

0 on Success or Error Code on Failure

Example:

```
ArmExitDev ( );
```

Before terminating the application, the supplied cleanup function should be called. ArmExitDev() closes the GPIB device driver. After this cleanup has been performed the application may terminate normally.

Download Setup Parameters to Arm Generator

```
long ArmSetParm ( AGEN *tAgen );
```

Input:

tAgen Pointer to Arm Generator Parameter Structure

Return:

0 on Success or Error Code on Failure

Example:

```
ArmSetParm ( &tAgen );
```

This function may be called to download the complete set of arm generator parameters. Note that the Arm Generator Parameter Structure contains all the information necessary to completely define an output state.

In order to optimize performance, this function keeps track of parameters that have been configured and only downloads parameters that have changed since the last time it was called. However, parameters which are manually sent using the ArmTalkDev() function will not be tracked, and could therefore cause unpredictable results. If this function is used to configure parameters, it should be used exclusively, and no parameters should be manually sent.

Fill an Arm Generator Structure with Default Values

```
void ArmDefParm ( AGEN *tAgen );
```

Input:

tAgen Pointer to Arm Generator Parameter Structure

Return:

None

Example:

```
AGEN tAgen;  
ArmDefAgen ( &tAgen );
```

This function is used to fill an Arm Generator Parameter Structure with default values. These parameters could then be downloaded to the AG-100 by calling the ArmSetParm() function. Using this function insures that all parameters contain reasonable values. It is not necessary to clear a Parameter Structure using the standard memset() function prior to calling this function, as no dynamic memory allocation information is contained within the Parameter Structure.

Arm Generator Reset

```
long ArmRsetDev ( void );
```

Input:

None

Return:

0 on Success or Error Code on Failure

Example:

```
ArmRsetDev ( );
```

This function will reset the device to the power-up state. The existing machine state is lost, and all parameters are restored to their default values.

Send Command String to Device

```
long ArmTalkDev ( char *sCmd );
```

Input:

sCmd Pointer to Command String

Return:

0 on Success or Error Code on Failure

Example:

```
ArmTalkDev ( ":PATT C14FAC14FA" );
```

This function may be used to send individual command strings to the device. This function should be used whenever no response is expected from the device.

Send Command String and Receive ASCII Response

```
long ArmRqstAsc ( char *sCmd, char *sVal, long lLeng );
```

Input:

sCmd Pointer to Command String
sVal Pointer to Buffer to Hold Response String
lLeng Length of Buffer to Hold Response String

Return:

0 on Success or Error Code on Failure
Response is placed in Response Buffer on Success

Example:

```
char buffer[128];  
ArmRqstAsc ( ":PATT?", buffer, 128 );
```

This function may be used to send individual command strings to the device when an ASCII response is expected.

Find Arm Delay for Optimal Marker Placement

```
long ArmFindDly ( AGEN *tAgen );
```

Input:

tAgen Pointer to Arm Generator Parameter Structure

Return:

0 on Success or Error Code on Failure

Example:

```
ArmFindDly ( &tAgen );
```

This function may be called to find the delay that provides the optimal marker placement. The settings contained in the **tParm** member of the AGEN structure are DTS207x parameters used as feedback for assessing the marker placement. When this function successfully returns, the **ICyclDly** and **IFineDly** parameters will be altered to the values that were determined to provide the greatest jitter tolerance.

WAVECREST CORPORATION

World Headquarters:

7626 Golden Triangle Drive
Eden Prairie, MN 55344
(952) 831-0030
FAX: (952) 831-4474
Toll Free: 1-800-733-7128
www.wavecrest.com

West Coast Office:

1735 Technology Drive, Ste. 400
San Jose, CA 95110
(408) 436-9000
FAX: (408) 436-9001
1-800-821-2272

Europe Office:

Lilienthalalle 25
D-80939 Munchen
011-49-89-32225330
FAX: 011-49-89-32225333

Japan Office:

2-1 Ohtemachi 2-Chome
Chyoda-Ku, Tokyo
100-0004, Japan
81-3-5960-5770