Wide Band Artificial Pulsar

Generated by Doxygen 1.6.1

### CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.2.3 update</td>
<td>9</td>
</tr>
<tr>
<td>3.4 Pulsar Class Reference</td>
<td>10</td>
</tr>
<tr>
<td>3.4.1 Detailed Description</td>
<td>10</td>
</tr>
<tr>
<td>3.4.2 Member Function Documentation</td>
<td>11</td>
</tr>
<tr>
<td>3.4.2.1 addPulse</td>
<td>11</td>
</tr>
<tr>
<td>3.4.2.2 delPulse</td>
<td>11</td>
</tr>
<tr>
<td>3.4.2.3 getPeriod</td>
<td>11</td>
</tr>
<tr>
<td>3.4.2.4 getPulse</td>
<td>11</td>
</tr>
<tr>
<td>3.4.2.5 getSignalAtTime</td>
<td>11</td>
</tr>
<tr>
<td>3.4.2.6 getTotalPhase</td>
<td>11</td>
</tr>
<tr>
<td>3.4.2.7 getUpdateInterval</td>
<td>12</td>
</tr>
<tr>
<td>3.4.2.8 printParams</td>
<td>12</td>
</tr>
<tr>
<td>3.4.2.9 renderProfile</td>
<td>12</td>
</tr>
<tr>
<td>3.4.2.10 setPeriod</td>
<td>12</td>
</tr>
<tr>
<td>3.4.2.11 setTotalPhase</td>
<td>12</td>
</tr>
<tr>
<td>3.4.2.12 setUpdateInterval</td>
<td>12</td>
</tr>
<tr>
<td>3.5 PulseParam Struct Reference</td>
<td>13</td>
</tr>
<tr>
<td>3.5.1 Detailed Description</td>
<td>13</td>
</tr>
<tr>
<td>3.6 SPI Class Reference</td>
<td>14</td>
</tr>
<tr>
<td>3.6.1 Member Function Documentation</td>
<td>14</td>
</tr>
<tr>
<td>3.6.1.1 begin_tr</td>
<td>14</td>
</tr>
<tr>
<td>3.6.1.2 init</td>
<td>14</td>
</tr>
<tr>
<td>3.6.1.3 wait</td>
<td>14</td>
</tr>
<tr>
<td>3.7 SPI_MOD5270 Class Reference</td>
<td>15</td>
</tr>
<tr>
<td>3.7.1 Member Function Documentation</td>
<td>15</td>
</tr>
<tr>
<td>3.7.1.1 begin_tr</td>
<td>15</td>
</tr>
<tr>
<td>3.7.1.2 init</td>
<td>15</td>
</tr>
<tr>
<td>3.7.1.3 wait</td>
<td>15</td>
</tr>
<tr>
<td>3.8 TelnetCntlr Class Reference</td>
<td>16</td>
</tr>
<tr>
<td>3.8.1 Member Function Documentation</td>
<td>16</td>
</tr>
<tr>
<td>3.8.1.1 acceptConnection</td>
<td>16</td>
</tr>
<tr>
<td>3.8.1.2 hasConnection</td>
<td>16</td>
</tr>
<tr>
<td>3.9 TelnetCntlr_MOD5270 Class Reference</td>
<td>17</td>
</tr>
<tr>
<td>3.9.1 Member Function Documentation</td>
<td>17</td>
</tr>
<tr>
<td>3.9.1.1 acceptConnection</td>
<td>17</td>
</tr>
<tr>
<td>3.9.1.2 displayPrompt</td>
<td>17</td>
</tr>
</tbody>
</table>
3.9.1.3 hasConnection .................................................. 17
3.9.1.4 update ............................................................ 17

3.10 Timer Class Reference ........................................... 19
  3.10.1 Detailed Description ......................................... 19
  3.10.2 Member Function Documentation ............................ 19
    3.10.2.1 getTime .................................................. 19
    3.10.2.2 getTimeInterval ....................................... 20
    3.10.2.3 setTimeInterval ..................................... 20
    3.10.2.4 sleep ................................................... 20
    3.10.2.5 startTimer ............................................ 20
    3.10.2.6 stopTimerAndWait ..................................... 20

3.11 Timer_MOD5270 Class Reference ............................... 21
  3.11.1 Member Function Documentation ............................ 21
    3.11.1.1 getTime ................................................ 21
    3.11.1.2 sleep .................................................. 21
    3.11.1.3 startTimer ............................................ 21
    3.11.1.4 stopTimerAndWait ..................................... 21

3.12 VGA Class Reference ............................................. 22
  3.12.1 Detailed Description ......................................... 22
  3.12.2 Member Function Documentation ............................ 22
    3.12.2.1 getGaindB .............................................. 22
    3.12.2.2 getSPIDriver ......................................... 23
    3.12.2.3 maxGaindB ............................................ 23
    3.12.2.4 minGaindB ............................................ 23
    3.12.2.5 numberOfAmps ......................................... 23
    3.12.2.6 setGaindB ............................................. 23
    3.12.2.7 setSPIDriver ........................................ 23
    3.12.2.8 update ............................................... 24

3.13 VGA_MAX2064 Class Reference .................................. 25
  3.13.1 Detailed Description ......................................... 25
  3.13.2 Member Function Documentation ............................ 25
    3.13.2.1 getGaindB .............................................. 25
    3.13.2.2 maxGaindB ............................................ 25
    3.13.2.3 minGaindB ............................................ 25
    3.13.2.4 numberOfAmps ......................................... 26
    3.13.2.5 setGaindB ............................................. 26
Chapter 1

Class Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- Filter ................................................................. 8
- Filter_MOD5270 .................................................. 9
- Pulsar ............................................................... 10
- PulseParam ......................................................... 13
- SPI ................................................................. 14
- SPI_MOD5270 ...................................................... 15
- Timer ............................................................... 19
- Timer_MOD5270 .................................................. 21
- VGA ................................................................. 22
- VGA_MAX2064 .................................................... 25
- WBAPCmd ......................................................... 27
- WBAPCntlr ......................................................... 28
- CmdCntlr ........................................................... 5
- TelnetCntlr ......................................................... 16
- TelnetCntlr_MOD5270 .......................................... 17
- WBAPMngr ......................................................... 30
- WebCntlr .......................................................... 34
- WebCntlr_MOD5270 ............................................. 35
- WBAPState ......................................................... 31
Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<table>
<thead>
<tr>
<th>Class</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CmdCntlr</td>
<td>5</td>
</tr>
<tr>
<td>Filter</td>
<td>8</td>
</tr>
<tr>
<td>Filter MOD5270</td>
<td>9</td>
</tr>
<tr>
<td>Pulsar</td>
<td>10</td>
</tr>
<tr>
<td>PulseParam</td>
<td>13</td>
</tr>
<tr>
<td>SPI</td>
<td>14</td>
</tr>
<tr>
<td>SPI MOD5270</td>
<td>15</td>
</tr>
<tr>
<td>TelnetCntlr</td>
<td>16</td>
</tr>
<tr>
<td>TelnetCntlr MOD5270</td>
<td>17</td>
</tr>
<tr>
<td>Timer</td>
<td>19</td>
</tr>
<tr>
<td>Timer MOD5270</td>
<td>21</td>
</tr>
<tr>
<td>VGA</td>
<td>22</td>
</tr>
<tr>
<td>VGA MAX2064</td>
<td>25</td>
</tr>
<tr>
<td>WBAPCmd</td>
<td>27</td>
</tr>
<tr>
<td>WBAPCntlr</td>
<td>28</td>
</tr>
<tr>
<td>WBAPMngr</td>
<td>30</td>
</tr>
<tr>
<td>WBAPState</td>
<td>31</td>
</tr>
<tr>
<td>WebCntlr</td>
<td>34</td>
</tr>
<tr>
<td>WebCntlr MOD5270</td>
<td>35</td>
</tr>
</tbody>
</table>
Chapter 3

Class Documentation

3.1 CmdCntlr Class Reference

#include <cmd_cntlr.hpp>

Inheritance diagram for CmdCntlr:

\[
\begin{align*}
\text{WBAPCntlr} & \quad \text{CmdCntlr} \\
\text{TelnetCntlr} & \quad \text{TelnetCntlr} \\
\text{TelnetCntlr} & \quad \text{TelnetCntlr}\_MOD5270
\end{align*}
\]

Public Member Functions

- virtual void reset ()
- virtual void setPrompt (const char *prompt)
- virtual void displayPrompt ()
- virtual void setInputFile (FILE *)
- virtual FILE * getInputFile ()
- virtual void update ()

Protected Member Functions

- virtual bool readCmd ()
- virtual bool parseCmd ()

Protected Attributes

- char cmd_buff [CMD_BUFFER_SIZE]

  Storage for a command line.
• unsigned argc
  The number of arguments in the command.

• char * argv [CMD_BUFFER_SIZE]
  Pointer to each arg in cmd_buff.

• const char * prompt
  The display prompt.

• FILE * input_f
  The file to read commands from.

### 3.1.1 Detailed Description

Updates the WBAP state via console commands from the user

### 3.1.2 Member Function Documentation

#### 3.1.2.1 void CmdCntlr::displayPrompt () [virtual]

Print the prompt text to the terminal
Reimplemented in TelnetCntlr_MOD5270.

#### 3.1.2.2 virtual FILE* CmdCntlr::getInputFile () [inline, virtual]

Get the input stream being used to read commands

**Returns:**

std::istream The input stream being read line-by-line

#### 3.1.2.3 bool CmdCntlr::parseCmd () [protected, virtual]

Parse the cmd buffer

**Returns:**

bool True if the syntax is correct

#### 3.1.2.4 bool CmdCntlr::readCmd () [protected, virtual]

Read chars into the cmd buffer

**Returns:**

bool True if the cmd buffer is ready to be parsed
3.1 CmdCntlr Class Reference

3.1.2.5 void CmdCntlr::reset () [virtual]

Clear the command buffer

3.1.2.6 virtual void CmdCntlr::setInputFile (FILE * input_f) [inline, virtual]

Set the input file to read commands from

Parameters:

input_f The input stream to be read line-by-line

3.1.2.7 virtual void CmdCntlr::setPrompt (const char * prompt) [inline, virtual]

Set the display text for the cmd prompt

Parameters:

prompt The command prompt text (constant, null-term. string)

3.1.2.8 void CmdCntlr::update () [virtual]

Read a line from the input stream, parse it, and execute the command

Reimplemented from WBAPCntlr.

Reimplemented in TelnetCntlr_MOD5270.

The documentation for this class was generated from the following files:

• cmd_cntlr.hpp
• cmd_cntlr.cpp
3.2 Filter Class Reference

#include <filter.hpp>

Inheritance diagram for Filter::

```
Filter
 |    Filter_MOD5270
```

Public Member Functions

- virtual void init ()
- virtual void setFilterMode (filter_mode mode, unsigned channel)
- virtual void update ()

3.2.1 Detailed Description

Handles the control of the GPIO interface to select the filter mode for each channel

3.2.2 Member Function Documentation

3.2.2.1 virtual void Filter::init () [inline, virtual]

Initialize GPIO interface
Reimplemented in Filter_MOD5270.

3.2.2.2 virtual void Filter::setFilterMode (filter_mode mode, unsigned channel) [inline, virtual]

Set the filter mode for a particular channel

Parameters:

- mode The desired filter mode
- channel The desired channel for the filter

Reimplemented in Filter_MOD5270.

3.2.2.3 virtual void Filter::update () [inline, virtual]

Update the GPIO pins
Reimplemented in Filter_MOD5270.

The documentation for this class was generated from the following file:

- filter.hpp
#include <filter_mod5270.hpp>

Inheritance diagram for Filter_MOD5270:

```
Filter

Filter_MOD5270
```

## Public Member Functions

- `void init ()`
- `void setFilterMode (filter_mode mode, unsigned channel)`
- `void update ()`

## 3.3.1 Detailed Description

The filter selector class for the NetBurner MOD5270

## 3.3.2 Member Function Documentation

### 3.3.2.1 void Filter_MOD5270::init () [virtual]

Initialize GPIO interface

Reimplemented from Filter.

### 3.3.2.2 void Filter_MOD5270::setFilterMode (filter_mode mode, unsigned channel) [virtual]

Set the filter mode for a particular channel

**Parameters:**

- `mode` The desired filter mode
- `channel` The desired channel for the filter

Reimplemented from Filter.

### 3.3.2.3 void Filter_MOD5270::update () [virtual]

Update the GPIO pins

Reimplemented from Filter.

The documentation for this class was generated from the following files:

- `filter_mod5270.hpp`
- `filter_mod5270.cpp`
3.4 Pulsar Class Reference

#include <pulsar.hpp>

Public Member Functions

• int addPulse (PulseParam &pp, int id)
• const PulseParam ∗ getPulse (unsigned id)
• void delPulse (int id)
• void setPeriod (float period)
• float getPeriod ()
• void setTotalPhase (float phase)
• float getTotalPhase ()
• void setUpdateInterval (float dt)
• float getUpdateInterval ()
• float getSignalAtTime (float t)
• void printParams ()

Protected Member Functions

• bool renderProfile ()

Protected Attributes

• std::map<int, PulseParam> pulses
  Pulses to generate.
• int next_pulse_id
  Next unused pulse ID.
• float period
  The period of all pulses.
• float total_phase
  The overall phase shift.
• float update_interval
  The update interval.
• std::vector<float> points
  The sampled points from the profile.
• bool needs_render
  The profile needs to be re-sampled.

3.4.1 Detailed Description

This class generates a pulsar signal with the provided pulse parameters
3.4 Pulsar Class Reference

3.4.2 Member Function Documentation

3.4.2.1 int Pulsar::addPulse (PulseParam & pp, int id)

Add a periodic pulse to the pulsar signal. Returns an assigned id so the pulse can be overwritten. Use id=-1 to add a new pulse.

3.4.2.2 void Pulsar::delPulse (int id)

Delete a pulse Use id=-1 to delete all pulses

3.4.2.3 float Pulsar::getPeriod () [inline]

Get the period of the pulsar

Returns:

The period in seconds

3.4.2.4 const PulseParam * Pulsar::getPulse (unsigned id)

Get the pulsar parameters for the specified pulse id

Parameters:

id The pulse ID

3.4.2.5 float Pulsar::getSignalAtTime (float t)

Returns the sum of all pulse signals at time t

Parameters:

t The time input for the function in seconds

Returns:

The output of the pulse function

3.4.2.6 float Pulsar:: getTotalPhase () [inline]

Get the total phase of the signal

Returns:

The total phase shift
3.4.2.7 float Pulsar::getUpdateInterval () [inline]

Get the update interval

Returns:

The time between updates in seconds

3.4.2.8 void Pulsar::printParams ()

Prints the current parameters of the pulsar

3.4.2.9 bool Pulsar::renderProfile () [protected]

Render the pulse profile

Returns:

True if the pulse rendered successfully

3.4.2.10 void Pulsar::setPeriod (float period) [inline]

Set the period of the pulsar

Parameters:

period The period in seconds

3.4.2.11 void Pulsar::setTotalPhase (float phase) [inline]

Set the total phase of the entire signal

Parameters:

phase The total phase shift applied to all pulses

3.4.2.12 void Pulsar::setUpdateInterval (float dt) [inline]

Set the update interval for the pulse steps

Parameters:

dt The time between updates in seconds

The documentation for this class was generated from the following files:

- pulsar.hpp
- pulsar.cpp
3.5 PulseParam Struct Reference

`#include <pulsar.hpp>`

**Public Attributes**

- float `width`
  
  *The FWHM of the pulse.*

- float `phase`
  
  *The relative phase of the pulse.*

- float `peak_amp`
  
  *The expected value (peak amplitude) of the gaussian pulse.*

### 3.5.1 Detailed Description

Contains parameters which describe a gaussian pulse signal

The documentation for this struct was generated from the following file:

- pulsar.hpp
3.6 SPI Class Reference

Inheritance diagram for SPI::

```
  SPI
    SPI_MOD5270
```

Public Member Functions

- virtual void init ()
- virtual void begin_tr (unsigned char *tbuff, volatile unsigned char *rbuff, unsigned length, unsigned cs)
- virtual void wait ()

3.6.1 Member Function Documentation

3.6.1.1 virtual void SPI::begin_tr (unsigned char *tbuff, volatile unsigned char *rbuff, unsigned length, unsigned cs) [inline, virtual]

Send and receive unsigned char buffer with given length

Parameters:

- `tbuff` The outgoing transmission buffer
- `rbuff` The incoming receive buffer
- `length` The maximum length for the receive buffer
- `cs` The SPI chip ID to send and receive from

Reimplemented in SPI_MOD5270.

3.6.1.2 virtual void SPI::init () [inline, virtual]

Initialize the SPI driver

Reimplemented in SPI_MOD5270.

3.6.1.3 virtual void SPI::wait () [inline, virtual]

Wait for buffered TR to complete

Reimplemented in SPI_MOD5270.

The documentation for this class was generated from the following file:

- spi.hpp
3.7 SPI_MOD5270 Class Reference

Inheritance diagram for SPI_MOD5270::

```
    SPI
   /   |
SPI_MOD5270
```

Public Member Functions

- void init ()
- void begin_tr (unsigned char *tbuff, volatile unsigned char *rbuff, unsigned length, unsigned cs)
- void wait ()

3.7.1 Member Function Documentation

3.7.1.1 void SPI_MOD5270::begin_tr (unsigned char * tbuff, volatile unsigned char * rbuff, unsigned length, unsigned cs) [virtual]

Send and receive unsigned char buffer with given length

Parameters:

- `tbuff` The outgoing transmission buffer
- `rbuff` The incoming receive buffer
- `length` The maximum length for the receive buffer
- `cs` The SPI chip ID to send and receive from

Reimplemented from SPI.

3.7.1.2 void SPI_MOD5270::init () [virtual]

Initialize the SPI driver

Reimplemented from SPI.

3.7.1.3 void SPI_MOD5270::wait () [virtual]

Wait for buffered TR to complete

Reimplemented from SPI.

The documentation for this class was generated from the following files:

- spi_mod5270_driver.hpp
- spi_mod5270_driver.cpp
3.8 TelnetCntlr Class Reference

Inheritance diagram for TelnetCntlr:

```
WBAPCntlr

CmdCntlr

TelnetCntlr

TelnetCntlr_MOD5270
```

Public Member Functions

- virtual void acceptConnection ()
- virtual bool hasConnection ()

3.8.1 Member Function Documentation

3.8.1.1 virtual void TelnetCntlr::acceptConnection () [inline, virtual]

Wait for TCP telnet connection on port 23 and process commands
Reimplemented in TelnetCntlr_MOD5270.

3.8.1.2 virtual bool TelnetCntlr::hasConnection () [inline, virtual]

Returns:

true if there is an active telnet connection

Reimplemented in TelnetCntlr_MOD5270.

The documentation for this class was generated from the following file:

- telnet_cntlr.hpp
3.9 TelnetCntlr_MOD5270 Class Reference

Inheritance diagram for TelnetCntlr_MOD5270::

```
WBAPCntlr

CmdCntlr

TelnetCntlr

TelnetCntlr_MOD5270
```

Public Member Functions

- virtual void displayPrompt ()
- virtual void update ()
- virtual void acceptConnection ()
- virtual bool hasConnection ()

3.9.1 Member Function Documentation

3.9.1.1 void TelnetCntlr_MOD5270::acceptConnection () [virtual]

Wait for TCP telnet connection on port 23 and process commands
Reimplemented from TelnetCntlr.

3.9.1.2 void TelnetCntlr_MOD5270::displayPrompt () [virtual]

Print the prompt text to the terminal
Reimplemented from CmdCntlr.

3.9.1.3 bool TelnetCntlr_MOD5270::hasConnection () [virtual]

Returns:
    true if there is an active telnet connection

Reimplemented from TelnetCntlr.

3.9.1.4 void TelnetCntlr_MOD5270::update () [virtual]

Read a line from the input stream, parse it, and execute the command
Reimplemented from CmdCntlr.

The documentation for this class was generated from the following files:
• telnet_cntlr_mod5270.hpp
• telnet_cntlr_mod5270.cpp
#include `<timing.hpp>`

Inheritance diagram for Timer:

```
{Timer

Timer_MOD5270
```

## Public Member Functions

- virtual void `startTimer()` = 0
- virtual float `getTime()` = 0
- virtual float `stopTimerAndWait()`
- virtual void `setTimeInterval(float time_interval)`
- virtual float `getTimeInterval()`
- virtual void `sleep(float s)` = 0

## Protected Attributes

- float `time_interval`
  
  *The desired time interval in seconds.*

- float `start_time`
  
  *The start time in seconds.*

## 3.10.1 Detailed Description

The timer class is responsible for maintaining a constant update rate for a loop.

Use:

- Before the loop call `setTimeInterval(float)` to set the desired update rate in seconds.
- At the start of the loop call `startTimer()`
- At the end of the loop call `stopTimerAndWait()` so that the timer can make the program sleep for the rest of the time interval.

If more time has passed than desired, the timer will not sleep.

## 3.10.2 Member Function Documentation

### 3.10.2.1 virtual float Timer::`getTime()` [pure virtual]

Get the current elapsed time in seconds

Implemented in `Timer_MOD5270`. 

---

Generated on Wed Aug 17 10:12:25 2016 for Wide Band Artificial Pulsar by Doxygen
3.10.2.2 virtual float Timer::getTimeInterval () [inline, virtual]

Returns:

The set time interval

3.10.2.3 virtual void Timer::setTimeInterval (float time_interval) [inline, virtual]

Set the desired time interval. The timer will do its best to keep this rate constant in real time

Parameters:

\textit{time\_interval}  The time interval in seconds

3.10.2.4 virtual void Timer::sleep (float \textit{s}) [pure virtual]

Sleep for specified time in seconds

Implemented in Timer\_MOD5270.

3.10.2.5 virtual void Timer::startTimer () [pure virtual]

Tell the timer that this is when we started the loop

Implemented in Timer\_MOD5270.

3.10.2.6 float Timer::stopTimerAndWait () [virtual]

If less time than desired has passed, sleep for the remainder of the update interval

Returns:

The amount of time slept in seconds

Reimplemented in Timer\_MOD5270.

The documentation for this class was generated from the following files:

- timing.hpp
- timing.cpp
3.11 Timer_MOD5270 Class Reference

Inheritance diagram for Timer_MOD5270:

```
Timer
|     
| Timer_MOD5270
```

Public Member Functions

• void startTimer ()
• float getTime ()
• float stopTimerAndWait ()
• void sleep (float s)

3.11.1 Member Function Documentation

3.11.1.1 float Timer_MOD5270::getTime () [virtual]

Get the current elapsed time in seconds
Implements Timer.

3.11.1.2 void Timer_MOD5270::sleep (float s) [virtual]

Sleep for specified time in seconds
Implements Timer.

3.11.1.3 void Timer_MOD5270::startTimer () [virtual]

Tell the timer that this is when we started the loop
Implements Timer.

3.11.1.4 float Timer_MOD5270::stopTimerAndWait () [virtual]

If less time than desired has passed, sleep for the remaineder of the update interval

Returns:

The amount of time slept in seconds

Reimplemented from Timer.

The documentation for this class was generated from the following files:

• timing_mod5270.hpp
• timing_mod5270.cpp
3.12 VGA Class Reference

#include <vga.hpp>

Inheritance diagram for VGA::

```
VGA
VGA_MAX2064
```

Public Member Functions

- virtual void setGaindB (float gain_db, unsigned amp_id)=0
- virtual float getGaindB (unsigned amp_id)=0
- virtual unsigned numberOfAmps ()=0
- virtual float maxGaindB ()=0
- virtual float minGaindB ()=0
- virtual void setSPIDriver (SPI ∗ spi_driver, unsigned chip_id)
- virtual SPI ∗ getSPIDriver ()
- virtual void update ()

Protected Attributes

- SPI ∗ spi_driver
  
  The SPI driver used to control the VGA.

- unsigned spi_id
  
  The SPI chip select line for the VGA.

3.12.1 Detailed Description

The VGA class is responsible for setting the correct gain for a variable-gain amplifier.

3.12.2 Member Function Documentation

3.12.2.1 virtual float VGA::getGaindB (unsigned amp_id) [pure virtual]

Parameters:

- amp_id The index of the amplifier

Returns:

- The set gain for amp_id

Implemented in VGA_MAX2064.
3.12.2.2 virtual SPI* VGA::getSPIDriver () [inline, virtual]

Returns:
The SPI driver

3.12.2.3 virtual float VGA::maxGainDB () [pure virtual]

Returns:
The maximum gain in dB of the VGA

Implemented in VGA_MAX2064.

3.12.2.4 virtual float VGA::minGainDB () [pure virtual]

Returns:
The minimum gain in dB of the VGA

Implemented in VGA_MAX2064.

3.12.2.5 virtual unsigned VGA::numberOfAmps () [pure virtual]

Returns:
The number of amplifiers available on the VGA chip

Implemented in VGA_MAX2064.

3.12.2.6 virtual void VGA::setGainDB (float gain_db, unsigned amp_id) [pure virtual]

Set the desired gain in dB for amp_id

Parameters:

\textit{gain\_db} The desired gain in dB

\textit{amp\_id} For VGA chips with multiple amplifiers, select the amp by index

Implemented in VGA_MAX2064.

3.12.2.7 virtual void VGA::setSPIDriver (SPI* spi_driver, unsigned chip_id) [inline, virtual]

Parameters:

\textit{spi\_driver} The driver for the SPI controller used to send commands to the VGA

\textit{chip\_id} The SPI chip ID for the VGA
3.12.2.8 virtual void VGA::update () [inline, virtual]

Update the VGA output gains
Reimplemented in VGA_MAX2064.
The documentation for this class was generated from the following file:

* vga.hpp
3.13 VGA_MAX2064 Class Reference

#include <vga_max2064_driver.hpp>

Inheritance diagram for VGA_MAX2064:

```
VGA
    │
    │
VGA_MAX2064
```

Public Member Functions

- void setGainDB (float gain_db, unsigned amp_id)
- float getGainDB (unsigned amp_id)
- unsigned numberOfAmps ()
- float maxGainDB ()
- float minGainDB ()
- void update ()

3.13.1 Detailed Description

Variable-gain amplifier driver for the MAX2064 chip used on the WBAP rev B

3.13.2 Member Function Documentation

3.13.2.1 float VGA_MAX2064::getGainDB (unsigned amp_id) [virtual]

Parameters:

- `amp_id`  The index of the amplifier

Returns:

- The set gain for amp_id

Implements VGA.

3.13.2.2 float VGA_MAX2064::maxGainDB () [inline, virtual]

Returns:

- The maximum gain in dB of the VGA

Implements VGA.

3.13.2.3 float VGA_MAX2064::minGainDB () [inline, virtual]

Returns:

- The minimum gain in dB of the VGA

Implements VGA.
3.13.2.4 unsigned VGA_MAX2064::numberOfAmps () [inline, virtual]

Returns:
The number of amplifiers available on the VGA chip

Implements VGA.

3.13.2.5 void VGA_MAX2064::setGaindB (float gain_db, unsigned amp_id) [virtual]

Set the desired gain in dB for amp_id

Parameters:
   gain_db The desired gain in dB
   amp_id For VGA chips with multiple amplifiers, select the amp by index

Implements VGA.

3.13.2.6 void VGA_MAX2064::update () [virtual]

Update the VGA output gains
Reimplemented from VGA.

The documentation for this class was generated from the following files:

- vga_max2064_driver.hpp
- vga_max2064_driver.cpp
3.14 WBAPCmd Struct Reference

Public Attributes

- const char * cmd
  
  The function to handle the command.

- int(* cmd_hndlr)(unsigned argc, const char *argv[], WBAPState *state)
  
  String containing the help text for the help command.

3.14.1 Member Data Documentation

3.14.1.1 const char* WBAPCmd::cmd

The name of the command

The documentation for this struct was generated from the following file:

- wbap_cmd.hpp
## 3.15 WBAPCntlr Class Reference

```cpp
#include <wbap_cntlr.hpp>
```

### Inheritance Diagram for WBAPCntlr:

```
WBAPCntlr
```

```
<table>
<thead>
<tr>
<th>CmdCntlr</th>
<th>WBAPMngr</th>
<th>WebCntlr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TelnetCntlr</td>
<td></td>
<td>WebCntlr_MOD5270</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TelnetCntlr_MOD5270</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

### Public Member Functions

- `virtual void setWBAPState (WBAPState *state)`
- `virtual WBAPState * getWBAPState ()`
- `virtual void update ()`

### Protected Attributes

- `WBAPState * wbap_state`

  *The state object used to control the WBAP.*

### 3.15.1 Detailed Description

Classes which inherit the `WBAPCntlr` class are able to control the state of the WBAP.

### 3.15.2 Member Function Documentation

#### 3.15.2.1 virtual WBAPState * WBAPCntlr::getWBAPState () [inline, virtual]

Get the WBAP state object

**Returns:**

Pointer to the WBAP state object

#### 3.15.2.2 virtual void WBAPCntlr::setWBAPState (WBAPState * state) [inline, virtual]

Set the WBAP state object

**Parameters:**

- `state` The state object used to control the state of the WBAP
3.15.2.3  virtual void WBAPCntlr::update ()  [inline, virtual]

Read the WBAP state and update the hardware accordingly
Reimplemented in CmdCntlr, TelnetCntlr_MOD5270, and WebCntlr_MOD5270.
The documentation for this class was generated from the following file:

* wbap_cntlr.hpp
3.16 WBAPMngr Class Reference

```cpp
#include <wbap_cntlr.hpp>
```

Inheritance diagram for WBAPMngr:

```
WBAPCntlr
│
└───WBAPMngr
```

Public Member Functions

- void `run()`

3.16.1 Detailed Description

This class updates the hardware according to the **WBAPState**

3.16.2 Member Function Documentation

3.16.2.1 void WBAPMngr::run()

Start the update loop

The documentation for this class was generated from the following files:

- wbap_cntlr.hpp
- wbap_cntlr.cpp
Public Member Functions

- void setFilter (Filter *filter)
- Filter *getFilter ()
- void setPulsar (Pulsar *pulsar)
- Pulsar *getPulsar ()
- void setState (const wbap_run_state &state)
- wbap_run_state getState ()
- void setTimer (Timer *timer)
- Timer *getTimer ()
- void setVGADriver (VGA *vga_driver)
- VGA *getVGADriver ()
- int runWBAPCmd (unsigned argc, const char *argv[])

Protected Attributes

- wbap_run_state state
  The run state of the WBAP.
- Pulsar *pulsar
  The pulsar signal generator.
- Timer *timer
  The global system timer for controlling update interval.
- VGA *vga_driver
  The driver for the variable gain amp.
- Filter *filter
  The filter selector.

3.17.1 Detailed Description

This class is used to control the state of the WBAP

3.17.2 Member Function Documentation

3.17.2.1 Filter *WBAPState::getFilter () [inline]

Returns:
  The filter driver
3.17.2.2  **Pulsar**  WBAPState::getPulsar ()  [inline]

**Returns:**

The pulsar signal generator

3.17.2.3  **wbap_run_state**  WBAPState::getState ()  [inline]

**Returns:**

The running state of the WBAP

3.17.2.4  **Timer**  WBAPState::getTimer ()  [inline]

**Returns:**

The timer used to control the update rate

3.17.2.5  **VGA**  WBAPState::getVGADriver ()  [inline]

**Returns:**

The VGA driver used to control the hardware output signal

3.17.2.6  **int**  WBAPState::runWBAPCmd (unsigned argc, const char * argv[])  

**Read and execute a command for the WBAP**

**Returns:**

Non-zero if command failed to run

3.17.2.7  **void**  WBAPState::setFilter (Filter * filter)  [inline]

**Parameters:**

*filter*  The filter driver

3.17.2.8  **void**  WBAPState::setPulsar (Pulsar * pulsar)  [inline]

**Parameters:**

*pulsar*  The pulsar signal generator

3.17.2.9  **void**  WBAPState::setState (const wbap_run_state & state)  [inline]

**Parameters:**

*state*  The running state of the WBAP
void WBAPState::setTimer (Timer *timer) [inline]

Parameters:

timer  The timer used to control the update rate

void WBAPState::setVGADriver (VGA *vga_driver) [inline]

Parameters:

vga_driver  The VGA driver used to control the hardware output signal

The documentation for this class was generated from the following files:

- wbap_state.hpp
- wbap_state.cpp
3.18 WebCntlr Class Reference

#include <web_cntlr.hpp>

Inheritance diagram for WebCntlr:

```
WBAPCntlr

WebCntlr

WebCntlr_MOD5270
```

Public Member Functions

- virtual bool startHTTP ()

3.18.1 Detailed Description

This class uses the web interface to control the WBAP state

3.18.2 Member Function Documentation

3.18.2.1 virtual bool WebCntlr::startHTTP () [inline, virtual]

Start the HTTP web server.
Reimplemented in WebCntlr_MOD5270.

The documentation for this class was generated from the following file:

- web_cntlr.hpp
3.19 WebCntlr_MOD5270 Class Reference

Inheritance diagram for WebCntlr_MOD5270::

```
   WBAPCntlr
   WebCntlr
   WebCntlr_MOD5270
```

Public Member Functions

- bool startHTTP ()
- void update ()

3.19.1 Member Function Documentation

3.19.1.1 bool WebCntlr_MOD5270::startHTTP () [virtual]

Start the HTTP web server
Reimplemented from WebCntlr.

3.19.1.2 void WebCntlr_MOD5270::update () [virtual]

Read the WBAP state and update the hardware accordingly
Reimplemented from WBAPCntlr.

The documentation for this class was generated from the following files:

- web_cntlr_mod5270.hpp
- web_cntlr_mod5270.cpp
Index

acceptConnection
  TelnetCntlr, 16
  TelnetCntlr_MOD5270, 17
addPulse
  Pulsar, 11
begin_tr
  SPI, 14
  SPI_MOD5270, 15
cmd
  WBAPCmd, 27
CmdCntlr, 5
  displayPrompt, 6
  getInputFile, 6
  parseCmd, 6
  readCmd, 6
  reset, 6
  setInputFile, 7
  setPrompt, 7
  update, 7
delPulse
  Pulsar, 11
displayPrompt
  CmdCntlr, 6
  TelnetCntlr_MOD5270, 17
Filter, 8
  init, 8
  setFilterMode, 8
  update, 8
Filter_MOD5270, 9
  init, 9
  setFilterMode, 9
  update, 9
getFilter
  WBAPState, 31
getGaindB
  VGA, 23
  VGA_MAX2064, 25
getInputFile
  CmdCntlr, 6
getPeriod
  Pulsar, 11
getPulsar
  WBAPState, 31
getPulse
  Pulsar, 11
getSignalAtTime
  Pulsar, 11
getSPIDriver
  VGA, 22
getState
  WBAPState, 32
getTime
  Timer, 19
  Timer_MOD5270, 21
g getTimeInterval
  Timer, 19
g getTime
  WBAPState, 32
g getTotalPhase
  Pulsar, 11
g getUpdateInterval
  Pulsar, 11
g getVGADriver
  WBAPState, 32
g getWBAPState
  WBAPCntlr, 28
hasConnection
  TelnetCntlr, 16
  TelnetCntlr_MOD5270, 17
init
  Filter, 8
  Filter_MOD5270, 9
  SPI, 14
  SPI_MOD5270, 15
maxGaindB
  VGA, 23
  VGA_MAX2064, 25
minGaindB
  VGA, 23
  VGA_MAX2064, 25
numberOfAmps
  VGA, 23
  VGA_MAX2064, 25
INDEX 37

parseCmd
  CmdCntlr, 6
printParams
  Pulsar, 12
Pulsar, 10
  addPulse, 11
delPulse, 11
getPeriod, 11
getPulse, 11
getSignalAtTime, 11
getTotalPhase, 11
getUpdateInterval, 11
printParams, 12
renderProfile, 12
setPeriod, 12
setTotalPhase, 12
setUpdateInterval, 12
PulseParam, 13
readCmd
  CmdCntlr, 6
renderProfile
  Pulsar, 12
reset
  CmdCntlr, 6
run
  WBAPMngr, 30
runWBAPCmd
  WBAPState, 32

setFilter
  WBAPState, 32
setFilterMode
  Filter, 8
  Filter_MOD5270, 9
setGaindB
  VGA, 23
  VGA_MAX2064, 26
setInputFile
  CmdCntlr, 7
setPeriod
  Pulsar, 12
setPrompt
  CmdCntlr, 7
setPulsar
  WBAPState, 32
setSPIDriver
  VGA, 23
setState
  WBAPState, 32
setTimeInterval
  Timer, 20
setTimer
  WBAPState, 32

setTotalPhase
  Pulsar, 12
setUpdateInterval
  Pulsar, 12
setVGADriver
  WBAPState, 33
setWBAPState
  WBAPCntlr, 28

sleep
  Timer, 20
  Timer_MOD5270, 21
SPI, 14
  begin_tr, 14
  init, 14
  wait, 14
SPI_MOD5270, 15
  begin_tr, 15
  init, 15
  wait, 15

startHTTP
  WebCntlr, 34
  WebCntlr_MOD5270, 35

startTimer
  Timer, 20
  Timer_MOD5270, 21

stopTimerAndWait
  Timer, 20
  Timer_MOD5270, 21

TelnetCntlr, 16
  acceptConnection, 16
  hasConnection, 16
TelnetCntlr_MOD5270, 17
  acceptConnection, 17
  displayPrompt, 17
  hasConnection, 17
  update, 17
Timer, 19
  getTime, 19
  getTimeInterval, 19
  setTimeInterval, 20
  sleep, 20
  startTimer, 20
  stopTimerAndWait, 20
Timer_MOD5270, 21
  getTime, 21
  sleep, 21
  startTimer, 21
  stopTimerAndWait, 21

update
  CmdCntlr, 7
  Filter, 8
  Filter_MOD5270, 9

Generated on Wed Aug 17 10:12:25 2016 for Wide Band Artificial Pulsar by Doxygen
TelnetCntlr_MOD5270, 17
VGA, 23
VGA_MAX2064, 26
WBAPCntlr, 28
WebCntlr_MOD5270, 35

VGA, 22
getGaindB, 22
getSPIDriver, 22
maxGaindB, 23
minGaindB, 23
numberOfAmps, 23
setGaindB, 23
setSPIDriver, 23
update, 23

VGA_MAX2064, 25
getGaindB, 25
maxGaindB, 25
minGaindB, 25
numberOfAmps, 25
setGaindB, 26
update, 26

wait
SPI, 14
SPI_MOD5270, 15

WBAPCmd, 27
cmd, 27

WBAPCntlr, 28
getWBAPState, 28
setWBAPState, 28
update, 28

WBAPMngr, 30
run, 30

WBAPState, 31
getFilter, 31
getPulsar, 31
gestate, 32
getTimer, 32
getVGADriver, 32
runWBAPCmd, 32
setFilter, 32
setPulsar, 32
setState, 32
setTimer, 32
setVGADriver, 33

WebCntlr, 34
startHTTP, 34

WebCntlr_MOD5270, 35
startHTTP, 35
update, 35