Corso di Laurea in Ingegneria delle Telecomunicazioni

Corso di Misure Elettroniche per le Telecomunicazioni II

Progettazione e testing di un trasmettitore DVB-C con modulazione 32-QAM e banda 4 MHz

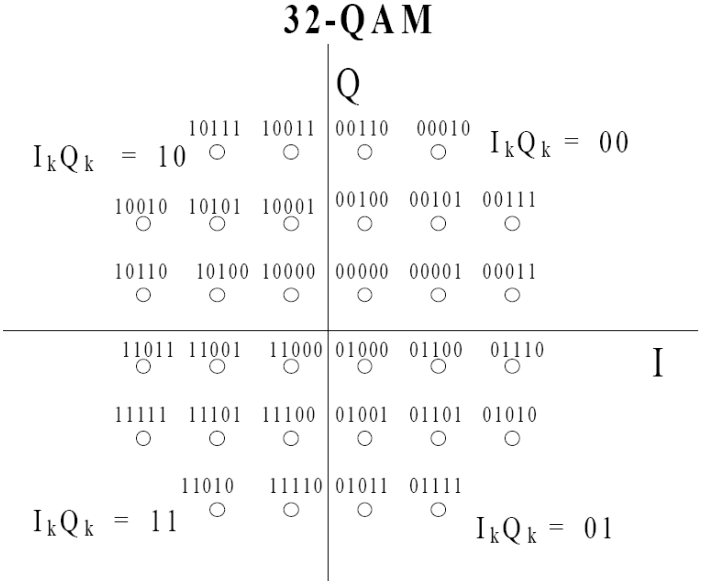
Emulare un segnale DVB-C con l'ausilio di un generatore arbitrario Agilent 33220A, avente le seguenti caratteristiche:

1. **fattore di roll-off α=0.15.**

Il DVB-C utilizza come filtro formatore un filtro radice di coseno rialzato, la cui formula analitica della risposta impulsiva è

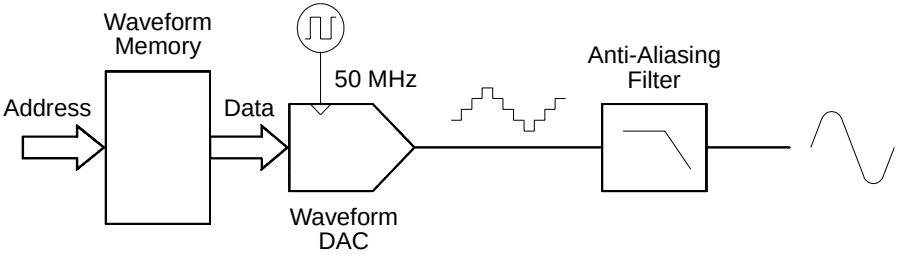
$$h(t)=\frac{\sin\left(\pi\frac{t}{T_S}(1-\alpha)\right)+4\alpha\frac{t}{T_S}\cos\left(\pi\frac{t}{T_S}(1+\alpha)\right)}{\pi\frac{t}{T_S}\left(1-\left(4\alpha\frac{t}{T_S}\right)^2\right)}$$

Effettuare N misure di larghezza di banda e restituire il valor medio delle misure e la deviazione standard.



Direct Digital Synthesis

The Agilent 33220A uses a signal-generation technique called *Direct Digital Synthesis* (DDS) for all waveform functions except pulse. As shown below, a stream of digital data representing the desired waveform is sequentially read from waveform memory and is applied to the input of a digital-to-analog converter (DAC). The DAC is clocked at the function generator's sampling frequency of 50 MHz and outputs a series of voltage steps approximating the desired waveform. A low-pass "anti-aliasing" filter then smooths the voltage steps to create the final waveform.

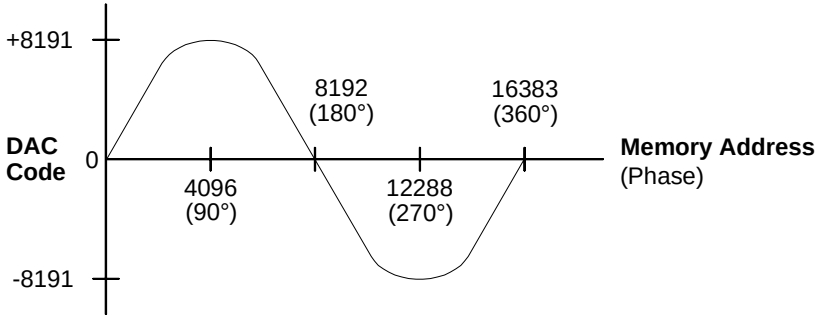


Direct Digital Synthesis Circuitry

The 33220A uses two anti-aliasing filters. An elliptical filter is used for continuous sine waves because of its nearly flat passband and sharp cutoff above 20 MHz. Since elliptical filters exhibit severe ringing for waveforms other than continuous sine waves, a linear-phase filter is used for all other waveform functions.

For standard waveforms, and arbitrary waveforms that are defined with fewer than 16,384 (16K) points, the function generator uses waveform memory that is 16K words deep. For arbitrary waveforms that are defined with more than 16K points, the function generator uses waveform memory that is 65,536 (64K) words deep.

The 33220A represents amplitude values by 16,384 discrete voltage levels (or 14-bit vertical resolution). The specified waveform data is divided into samples such that one waveform cycle exactly fills waveform memory (see the illustration below for a sine wave). If you create an arbitrary waveform that does not contain exactly 16K or 64K points, the waveform is automatically "stretched" by repeating points or by interpolating between existing points as needed to fill waveform memory. Since all of waveform memory is filled with one waveform cycle, each memory location corresponds to a phase angle of 2π/16,384 radians or 2π/65,536 radians.



Sine Wave Representation in Waveform Memory

Direct digital synthesis (DDS) generators use a *phase accumulation* technique to control waveform memory addressing. Instead of using a counter to generate sequential memory addresses, an "adder" is used (see the following page). On each clock cycle, the constant loaded into the phase increment register (PIR) is added to the present result in the phase accumulator. The most-significant bits of the phase accumulator output are used to address waveform memory. By changing the PIR constant, the number of clock cycles required to step through the entire waveform memory changes, thus changing the output frequency.

The PIR determines how fast the phase value changes with time and thereby controls the frequency being synthesized. More bits in the phase accumulator result in finer frequency resolution. Since the PIR affects only the rate of change of the phase value (and not the phase itself), changes in waveform frequency are phase-continuous.

Arbitrary Waveform Commands

(*see page 227 for more information)*

```
DATA VOLATILE, <value>, <value>, . . .

DATA:DAC VOLATILE, {<binary block>|<value>, <value>, . . . }

FORMat:BORDer {NORMal|SWAPped}        Specify Byte Order
FORMat:BORDer?

DATA:COPY <destination arb name> [,VOLATILE]

FUNCtion:USER {<arb name>[1]|VOLATILE}
FUNCtion:USER?

FUNCtion USER
FUNCtion?

DATA
  :CATalog?
  :NVOLatile:CATalog?
  :NVOLatile:FREE?

DATA:DELete <arb name>
DATA:DELete:ALL

DATA
  :ATTRibute:AVERage? [<arb name>[1]]
  :ATTRibute:CFACtor? [<arb name>[1]]
  :ATTRibute:POINts? [<arb name>[1]]
  :ATTRibute:PTPeak? [<arb name>[1]]
```

[1] *The names of the built-in arb waveforms are*: EXP_RISE, EXP_FALL, NEG_RAMP, SINC, and CARDIAC.

Parameters shown in **bold** *are selected following a* `*RST` *(reset) command.*

Arbitrary Waveform Commands

See also "Arbitrary Waveforms" starting on page 120 in chapter 3.

Arbitrary Waveform Overview

The following is an overview of the steps required to download and output an arbitrary waveform over the remote interface. The commands used for arbitrary waveforms are listed on page 229. *Refer to chapter 7, "Tutorial", for more information on the internal operation of downloading and outputting an arbitrary waveform.*

Note. *You can download waveforms of up to 65,536 (64K) points into the Agilent 33220A from your PC. However, waveforms of greater than 16,384 (16K) points cannot be edited from the Agilent 33220A front panel.*

Chapter 6, "Application Programs", includes an example program showing how to download an arbitrary waveform into the Agilent 33220A.

1 Download the waveform points into volatile memory.

You can download from 1 point (a dc signal) to 65,536 (64K) points per waveform. You can download the points as floating-point values, binary integer values, or decimal integer values. Use the `DATA` command to download floating-point values from -1.0 to +1.0. Use the `DATA:DAC` command to download binary integer or decimal integer values from -8191 to +8191.

To ensure that binary data is downloaded properly, you must select the order in which the bytes are downloaded using the `FORM:BORD` *command.*

2 Select the waveform frequency, amplitude, and offset.

Use the `APPLy` command or the equivalent `FREQ`, `VOLT`, and `VOLT:OFFS` commands to select the frequency, amplitude, and offset of the waveform.

3 Copy the arbitrary waveform to non-volatile memory.

You can output the arbitrary waveform directly from volatile memory or you can copy the waveform to non-volatile memory using the `DATA:COPY` command.

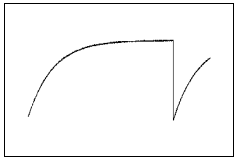
4 Select the arbitrary waveform to output.

You can select one of the five built-in arbitrary waveforms, one of four user-defined waveforms, or the waveform currently downloaded to volatile memory. Use the `FUNC:USER` command to select the waveform.

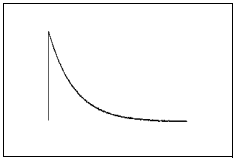
5 Output the selected arbitrary waveform.

Use the `FUNC USER` command to output the waveform previously selected with the `FUNC:USER` command.

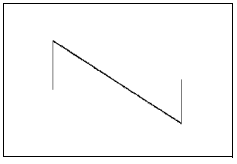
The five built-in arbitrary waveforms are shown below.



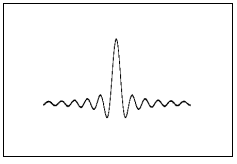
Exponential Rise



Exponential Fall



Negative Ramp



Sinc

Cardiac

Arbitrary Waveform Commands

DATA VOLATILE, <*value*>**,** <*value*>**, . . .**

Download *floating-point* values from -1 to +1 into volatile memory. You can download from 1 to 65,536 (64K) points per waveform. The function generator takes the specified number of points and expands them to fill waveform memory. If you download *less than* 16,384 (16K) points, a waveform with 16,384 points is automatically generated. If you download *more than* 16,384 points, a 65,536-point waveform is generated.

- The values -1 and +1 correspond to the *peak* values of the waveform (if the offset is 0 volts). For example, if you set the amplitude to 10 Vpp (0V offset), "+1" corresponds to +5V and "-1" corresponds to -5V.
- The maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "Sinc" waveform does not use the full range of values between ±1 and therefore its maximum amplitude is 6.087 Vpp (into 50 ohms).
- Downloading floating-point values (using `DATA VOLATILE`) is slower than downloading binary values (using `DATA:DAC VOLATILE`) but is more convenient when using trigonometric functions which return values from -1 to +1.
- The `DATA` command overwrites the previous waveform in volatile memory (and no error will be generated). Use the `DATA:COPY` command to copy the waveform to *non-volatile* memory.
- Up to four user-defined waveforms can be stored in non-volatile memory. Use the `DATA:DEL` command to delete the waveform in volatile memory or any of the four user-defined waveforms in non-volatile memory. Use the `DATA:CAT?` command to list all waveforms currently stored in volatile and non-volatile memory (as well as the five built-in waveforms).
- After downloading the waveform data to memory, use the `FUNC:USER` command to choose the active waveform and the `FUNC USER` command to output it.
- The following statement shows how to use the `DATA` command to download seven points to volatile memory.

```
DATA VOLATILE, 1, .67, .33, 0, -.33, -.67, -1
```

DATA:DAC VOLATILE, {<*binary block*>|<*value*>, <*value*>, . . . }

Download *binary* or *decimal* integer values from -8191 to +8191 into volatile memory. You can download from 1 to 65,536 (64K) points per waveform in IEEE-488.2 binary block format or as a list of values. The range of values corresponds to the values available using internal 14-bit DAC (Digital-to-Analog Converter) codes. The function generator takes the specified number of points and expands them to fill waveform memory. If you download *less than* 16,384 (16K) points, a waveform with 16,384 points is automatically generated. If you download *more than* 16,384 points, a 65,536-point waveform is generated.

- The values -8191 and +8191 correspond to the peak values of the waveform (if the offset is 0 volts). For example, if you set the output amplitude to 10 Vpp, "+8191" corresponds to +5V and "-8191" corresponds to -5V.
- The maximum amplitude will be limited if the data points do not span the full range of the output DAC. For example, the built-in "Sinc" waveform does not use the full range of values between ±8191 and therefore its maximum amplitude is 6.087 Vpp (into 50 ohms).
- The `DATA:DAC` command overwrites the previous waveform in volatile memory (and no error will be generated). Use the `DATA:COPY` command to copy the waveform to *non-volatile* memory.
- Up to four user-defined waveforms can be stored in non-volatile memory. Use the `DATA:DEL` command to delete the waveform in volatile memory or any of the four user-defined waveforms in non-volatile memory. Use the `DATA:CAT?` command to list all waveforms currently stored in volatile and non-volatile memory (as well as the five built-in waveforms).
- After downloading the waveform data to memory, use the `FUNC:USER` command to choose the active waveform and the `FUNC USER` command to output it.

- The following statement shows how to use the DATA:DAC command to download seven integer points using the binary block format (*see also "Using the IEEE-488.2 Binary Block Format" below*).

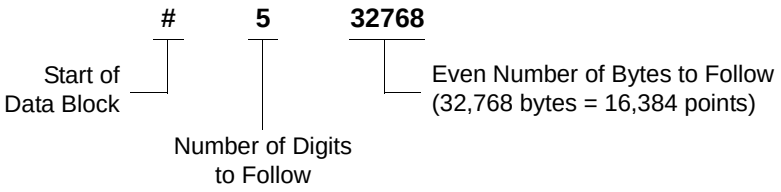
```
DATA:DAC VOLATILE, #214    Binary Data
```

- The following statement shows how to use the DATA:DAC command to download five integer points in decimal format.

```
DATA:DAC VOLATILE, 8191, 4096, 0, -4096, -8191
```

Using the IEEE-488.2 Binary Block Format

In the binary block format, a *block header* precedes the waveform data. The block header has the following format:



The function generator represents binary data as 16-bit integers, which are sent as two bytes. Therefore, **the total number of bytes is always twice the number of data points in the waveform** (and must always be an **even number**). For example, 32,768 bytes are required to download a waveform with 16,384 points.

Use the FORM:BORD command to select the byte order for binary transfers in block mode. If you specify FORM:BORD NORM (default), the most-significant byte (MSB) of each data point is assumed first. If you specify FORM:BORD SWAP, the least-significant byte (LSB) of each data point is assumed first. Most computers use the "swapped" byte order.

FORMat:BORDer {**NORMal**|**SWAPped**}
FORMat:BORDer?

Used for binary block transfers only. Select the byte order for binary transfers in the block mode using the DATA:DAC command. The default is NORM. The :BORD? query returns "NORM" or "SWAP".

- In *NORM* byte order (default), the most-significant byte (MSB) of each data point is assumed first.
- In *SWAP* byte order, the least-significant byte (LSB) of each data point is assumed first. Most computers use the "swapped" byte order.
- The function generator represents binary data as signed 16-bit integers, which are sent as two bytes. Therefore, each waveform data point requires 16 bits, which must be transferred as two bytes on the function generator's interfaces.

DATA:COPY <*destination arb name*> [**,VOLATILE**]

Copy the waveform from volatile memory to the specified name in non-volatile memory. The source for the copy is always "volatile". You cannot copy ***from*** any other source and you cannot copy ***to*** "volatile".

- The arb name may contain up to 12 characters. The first character *must* be a letter (A-Z), but the remaining characters can be numbers (0-9) or the underscore character (" _ "). Blank spaces are not allowed. If you specify a name with more than 12 characters, a "Program mnemonic too long" error is generated.
- The VOLATILE parameter is optional and can be omitted. Note that the keyword "VOLATILE" *does not* have a short form.
- The following built-in waveform names are reserved and cannot be used with the DATA:COPY command: "EXP_RISE", "EXP_FALL", "NEG_RAMP", "SINC", and "CARDIAC". If you specify one of the built-in waveforms, a "Cannot overwrite a built-in waveform" error is generated.
- The function generator does not distinguish between upper- and lower-case letters. Therefore, **ARB_1** and **arb_1** are the same name. All characters are converted to upper case.

- If you copy to a waveform name that already exists, the previous waveform is overwritten (and no error will be generated). However, you cannot overwrite any of the five built-in waveforms.
- Up to four user-defined waveforms can be stored in non-volatile memory. If memory is full and you try to copy a new waveform to non-volatile memory, a "Not enough memory" error is generated. Use the `DATA:DEL` command to delete the waveform in volatile memory or any of the four user-defined waveforms in non-volatile memory. Use the `DATA:CAT?` command to list all waveforms currently stored in volatile and non-volatile memory. The default selection is `EXP_RISE`.
- The following statement shows how to use the `DATA:COPY` command to copy the VOLATILE waveform into named storage "ARB_1".

```
DATA:COPY ARB_1, VOLATILE
```

FUNCtion:USER {<*arb name*>|**VOLATILE**}
FUNCtion:USER?

Select one of the five built-in arbitrary waveforms, one of four user-defined waveforms, or the waveform currently downloaded to volatile memory. The `:USER?` query returns "EXP_RISE", "EXP_FALL", "NEG_RAMP", "SINC", "CARDIAC", "VOLATILE", or the name of any user-defined waveforms in non-volatile memory. The default selection is "EXP_RISE".

- Note that this command *does not* output the selected arbitrary waveform. Use the `FUNC USER` command (see the following page) to output the selected waveform.
- The names of the five built-in arbitrary waveforms are: "EXP_RISE", "EXP_FALL", "NEG_RAMP", "SINC", and "CARDIAC".
- To select the waveform currently stored in volatile memory, specify the VOLATILE parameter. The keyword "VOLATILE" *does not* have a short form.
- If you select a waveform name that is not currently downloaded, a "Specified arb waveform does not exist" error is generated.

- The function generator does not distinguish between upper- and lower-case letters. Therefore, **ARB_1** and **arb_1** are the same name. All characters are converted to upper case.
- Use the DATA:CAT? command to list the names of the five built-in waveforms (non-volatile), "VOLATILE" if a waveform is currently downloaded to volatile memory, and the names of any user-defined waveforms (non-volatile).

FUNCtion USER
FUNCtion?

Select the arbitrary waveform function and output the current arbitrary waveform. When executed, this command outputs the arbitrary waveform currently selected by the FUNC:USER command (see the previous page). The selected waveform is output using the current frequency, amplitude, and offset voltage settings. The FUNC? query returns "SIN", "SQU", "RAMP", "PULS", "NOIS", "DC", or "USER".

- Use the APPLy command or the equivalent FREQ, VOLT, and VOLT:OFFS commands to select the frequency, amplitude, and offset of the waveform.
- The maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "SINC" waveform does not use the full range of binary values between ±1 and therefore its maximum amplitude is 6.087 Vpp (into 50 ohms).
- If you select an arbitrary waveform as the *modulating* waveshape ("USER"), the waveform is automatically limited to 4K points. Extra waveform points are removed using decimation.

DATA:CATalog?

List the names of *all* waveforms currently available for selection. Returns the names of the five built-in waveforms (non-volatile memory), "VOLATILE" if a waveform is currently downloaded to volatile memory, and all user-defined waveforms downloaded to non-volatile memory.

- A series of quoted strings separated with commas is returned as shown in the example below.

```
"VOLATILE","EXP_RISE","EXP_FALL","NEG_RAMP",
"SINC","CARDIAC","TEST1_ARB","TEST2_ARB"
```

- Use the DATA:DEL command to delete the waveform in volatile memory or any of the user-defined waveforms in non-volatile memory.

DATA:NVOLatile:CATalog?

List the names of all user-defined arbitrary waveforms downloaded to *non-volatile* memory. Returns the names of up to four waveforms.

- A series of quoted strings separated with commas is returned as shown in the example below. If no user-defined waveforms are currently downloaded, the command returns a null string (" ").

```
"TEST1_ARB","TEST2_ARB","TEST3_ARB","TEST4_ARB"
```

- Use the DATA:DEL command to delete any of the user-defined waveforms in non-volatile memory.

DATA:NVOLatile:FREE?

Query the number of non-volatile memory slots available to store user-defined waveforms. Returns the number of memory slots available to store user-defined waveforms. Returns "0" (memory is full), "1", "2", "3", or "4".

DATA:DELete <*arb name*>

Delete the specified arbitrary waveform from memory. You can delete the waveform in volatile memory or any of the four user-defined waveforms in non-volatile memory.

- You cannot delete the arbitrary waveform that is currently being output. If you attempt to delete this waveform, a "Not able to delete the currently selected active arb waveform" error is generated.
- You cannot delete any of the five built-in arbitrary waveforms. If you attempt to delete one of these waveforms, a "Not able to delete a built-in arb waveform" error is generated.
- Use the DATA:DEL:ALL command to delete the waveform in volatile memory and all user-defined non-volatile waveforms *all at once.* If one of the waveforms is currently being output, a "Not able to delete the currently selected active arb waveform" error is generated.

DATA:DELete:ALL

Delete all user-defined arbitrary waveforms from memory. This command deletes the waveform in volatile memory and all user-defined waveforms in non-volatile memory. The five built-in waveforms in non-volatile memory *are not* deleted.

- The colon before the ALL parameter is required (DATA:DELete:ALL). If you insert a space instead of a colon, the function generator will attempt to delete an arbitrary waveform with the name "ALL". If no such waveform is stored in memory, a "Specified arb waveform does not exist" error is generated.
- Use the DATA:DEL <*arb name*> command to delete stored waveforms *one at a time.*
- You cannot delete the arbitrary waveform that is currently being output. If you attempt to delete this waveform, a "Not able to delete the currently selected active arb waveform" error is generated.
- You cannot delete any of the five built-in arbitrary waveforms. If you attempt to delete one of these waveforms, a "Not able to delete a built-in arb waveform" error is generated.

DATA:ATTRibute:AVERage? [<*arb name*>]

Query the *arithmetic average* of all data points for the specified arbitrary waveform (-1 ≤ *average* ≤ +1). The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- If you query a waveform that is not currently stored in memory, a "Specified arb waveform does not exist" error is generated.

DATA:ATTRibute:CFACtor? [<*arb name*>]

Query the *crest factor* of all data points for the specified arbitrary waveform. Crest factor is the ratio of the peak value to the RMS value of the waveform. The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- If you query a waveform that is not currently stored in memory, a "Specified arb waveform does not exist" error is generated.

DATA:ATTRibute:POINts? [<*arb name*>]

Query the *number of points* in the specified arbitrary waveform. Returns a value from 1 to 65,536 points. The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- If you query a waveform that is not currently stored in memory, a "Specified arb waveform does not exist" error is generated.

DATA:ATTRibute:PTPeak? [<*arb name*>]

Query the *peak-to-peak* value of all data points for the specified arbitrary waveform. The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- This command returns a value from "0" to "+1.0", with "+1.0" indicating full amplitude available.
- The maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "Sinc" waveform does not use the full range of binary values between ±1 and therefore its maximum amplitude is 6.087 Vpp (into 50 ohms).
- If you query a waveform that is not currently stored in memory, a "Specified arb waveform does not exist" error is generated.