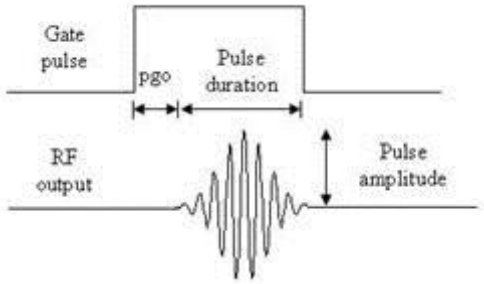
shapedrf

This command generates an RF pulse at the current frequency which can be modulated in amplitude and phase. It also includes a gate pulse which is used to control an RF amplifier.



Syntax (single pulse version)

```
shapedrf(destination, ampTable, phaseTable, phaseOffset, tableSize,
stepTime, [frequency])
```

`destination ...` "1" for channel 1 (Proton)
"2" for channel 2 (X-Channel)

`ampTable .....` the name of table which controls the amplitude during the generation of the shaped pulse. It has the form "tx" where x is an integer greater than 0. The table should contain positive 14 bit integers with a minimum value of 0 and a maximum of 16383. Use the command `ucsRun:convertTxGain()` to convert the amplitude in dB to one in this linear range

`phaseTable .....` the name of table which controls the phase of each element of the shaped RF pulse. It has the form "tx" where x is an integer greater than 0. The table should contain positive 16 bit integers with a minimum value of 0 and a maximum of 65535. This corresponds to phase shifts between 0 and 359.995 degrees.

`phaseOffset ...` the name of the parameter which controls the RF pulse phase relative to the system clock. It has the form "`px`" where `x` is an integer greater than 0. Phase values can be 0,1,2,3 corresponding to x, y, -x, –y (0, 90, 180, 270 degrees). Typically this is used to implement phase cycling. Fractional numbers can also be applied to implement other phases between 0 and 360 (0-4). The phase can also be supplied as a positive 16 bit integer in which case the parameter name has the form "nx" where x is an integer greater than 0. Constants can also be used.

`tableSize .....` the name of the parameter which controls the number of values in each of the amplitude and phase tables (both should have the same number of entries). It has the form "nx" where x is an integer greater than 0 or a constant can be supplied.

`stepTime .....` the name of the parameter which controls the duration of each amplitude step. It has the form "dx" where x is an integer greater than 0 or a constant can be used.

`frequency .....` the name of the optional parameter which controls frequency of the waveform. It has the form "fx" where x is an integer greater than 0 or a constant can be used. By default this will take the current channel 1 or 2 frequency. If not used ensure that the channel 2 frequency is set using the settxfreq or settxfreqs commands.

Notes:

1. The pulse generated by this command has a length which includes a preferences defined delay called the pulse gate overhead (pgo). This delay is used to set up the RF gate – a signal sent to the high powered amplifier to switch on the DC biasing before the RF pulse is actually sent. You should account for this delay when writing the relationships list described above (the predefined variable "pgo" (pulse gate overhead) can be used. The minimum pgo possible with this command is 1.5 μs however to ensure that you can work with dual pulses in other parts of the pulse program you should set this to 5 μs.

2. There are a maximum number of amplitude steps available determined by the amount of DSP memory available. The total for all amplitude and phase tables must be less than 128 k values.

3. The stepTime value may range from 2 μs to 327,670 μs. **Beware**: values below 2 μs can cause the RF to switch on for very long times.

4. Potentially the shaped RF pulse could have a large amplitude and last for a long time – the combination of these situation could damage a high-powered RF amplifier or probe. Always take care when setting the amplitude and setTime values. Place limits where possible.

5. To update the shaped RF pulse between scans requires resending the tables to the DSP however the phaseOffset parameter can be applied using a phase cycle table to adjust the phase by a constant amount based on the scan number.

6. In addition to the pgo delay at the beginning of the command it takes 0.35 μs for the command to finish before another command can be given. The dual pulse version takes xx μs. Consequently you should always place a delay of at least 1 μs between rf pulse commands.

7. The total duration (in μs) of this command will be `pgo + tableSize*stepTime`.

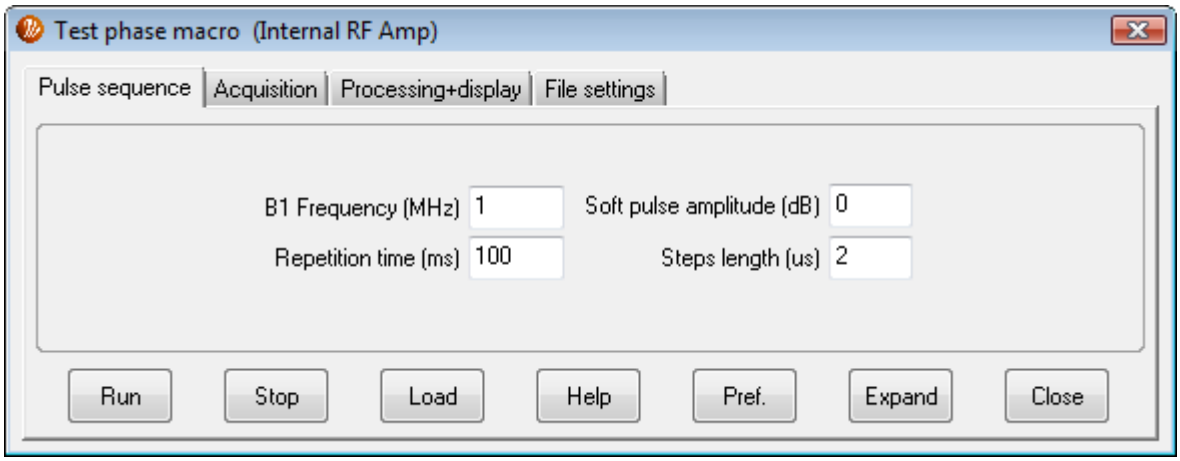
Example pulse sequence:

```
##########################################################
#
# Generate a ramped RF pulse with 5 steps each step
# 90 degrees shifted relative to the previous one.
# phase cycling will shift all phases by 180 degree
# each scan.
#
##########################################################

procedure(pulse_program,dir,mode)


# Interface description (name, label, x, y, ctrl, vartype)
  interface = ["b1Freq",     "B1 Frequency (MHz)",        "0","0", "tbw","freq",
               "repTime",    "Repetition time (ms)",      "0","1", "tbw","reptime",
               "spAmp",      "Soft pulse amplitude (dB)", "1","0", "tb","pulseamp",
               "stepLength", "Step length (us)",          "1","1", "tb","float,
[2,1000]"]

# Relationships to determine remaining variable values
   relationships = ["n1 = 5",
                    "d1 = stepLength",
                    "s = [0.2, 0.4, 0.6, 0.8, 1.0]",
                    "t1 = ucsRun:convertTxGain(spAmp)*mag(s)", # Must be positive
                    "t2 = [0, 90, 180, 270, 360]/360*2^16",
                    "n2 = nrPnts",
                    "totPnts = nrPnts",
                    "totTime = acqTime"]

# Define the tabs and their order
   tabs = ["Pulse sequence","Acquisition","Processing+Display","File Settings"]

# These parameters will be changed between experiments
   variables = [""]

# dx,dy
   dim = [180,26]

# Pulse sequence
   initpp(dir) # Reset internal parameter list
   shapedrf(1,t1,t2,p1,n1,d1)     # Shaped pulse on channel 1
   delay(10)                      # Pulse - acquire delay
   acquire("overwrite",n2)        # Acquire data

   lst = endpp() # Return parameter list

# Phase cycle list
   phaseList = [2,0; # Shaped pulse phase
                0,0] # Acquisition phase

endproc(lst,tabs,interface,relationships,variables,dim,phaseList)
```

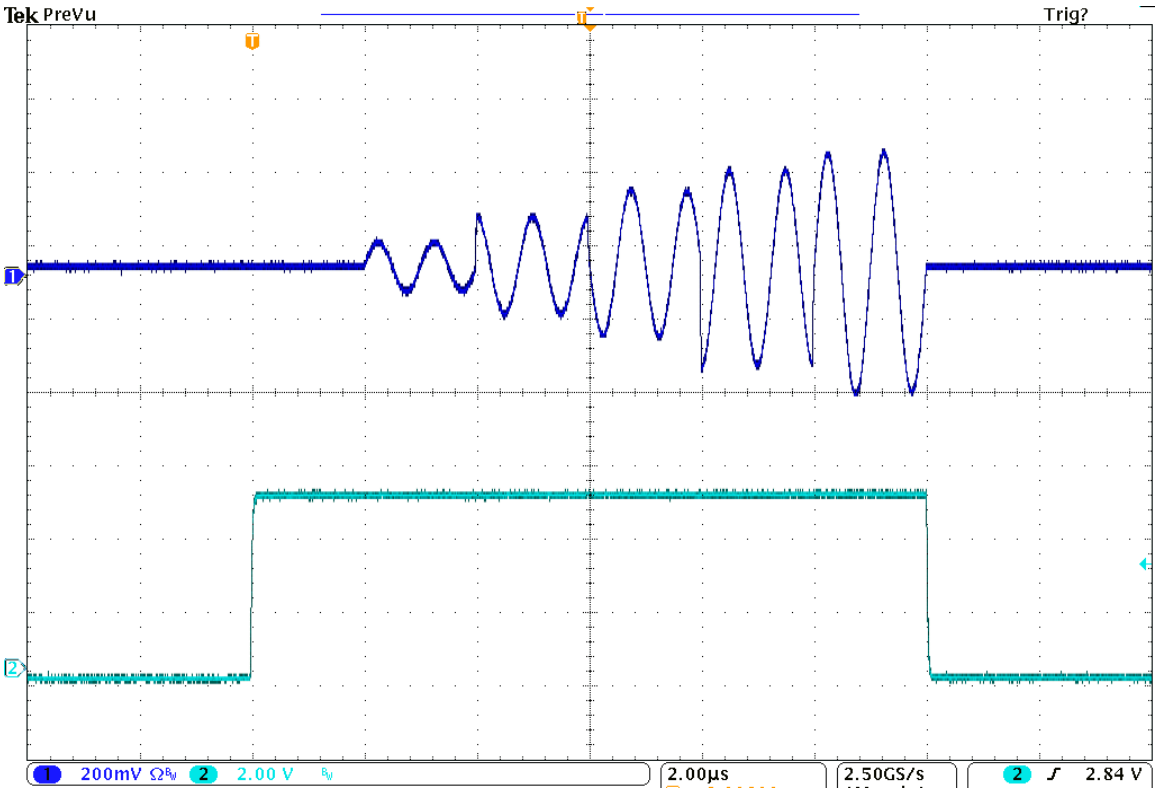
Note the line defining t1 – this is how the maximum amplitude of the soft pulse can be converted from dB to a positive 14 bit number. If negative amplitudes are required then phases of 180 degrees can be used. e.g.

```
s = [-1:0.1:1]
t1 = ucsRun:convertTxGain(spAmp)*mag(s)
t2 = (s<0)*180/360*2^16 # 180 if negative 0 if positive
```

Test parameters



Output, showing shaped RF pulse (blue) and TTL gate pulse (green) with pgo = 2 µs.



Syntax (dual pulse version)

```
shapedrf(ampTable, phaseTable, phaseOffset, frequency1, frequency2,
tableSize, stepTime)
```

`ampTable .....` the name of table which controls the amplitude during the generation of the shaped pulses. It has the form "tx" where x is an integer greater than 0. The table should contain positive 14 bit integers with a minimum value of 0 and a maximum of 16383. Use the command `ucsRun:convertTxGain()` to convert the amplitude in dB to one in this linear range. Note that the amplitude table is shared between the two pulses with the values alternating between the channels a_0^1, a_0^2, a_1^1, a_1^2, ... a_{n-1}^1, a_{n-1}^2 where the subscripts are the channel numbers. The easiest way to generate this is by combining two tables together using complex numbers

t_{Tot} = ctor(t1 + i*t2)

The conversion to complex numbers automatically interleaves the two tables, t1 and t2, while the complex to real command (ctor) converts this back to a real array but keeps the alternating format.

`phaseTable .....` the name of table which controls the phase of each element of the shaped RF pulse. It has the form "tx" where x is an integer greater than 0. The table should contain positive 16 bit integers with a minimum value of 0 and a maximum of 65535. This corresponds to phase shifts between 0 and 359.995 degrees. Like the amplitude table this must combine the two channel phase tables by interleaving them.

`phaseOffset ...` the name of the parameter which controls the RF pulse phase relative to the system clock. It has the form "`px`" where `x` is an integer greater than 0. Phase values can be 0,1,2,3 corresponding to x, y, -x, –y (0, 90, 180, 270 degrees). Typically this is used to implement phase cycling. Fractional numbers can also be applied to implement other phases between 0 and 360 (0-4). The phase can also be supplied as a positive 16 bit integer in which case the parameter name has the form "nx" where x is an integer greater than 0. Constants can also be used. This phase offset is applied to both tables.

`frequency1 .....` the name of the parameter which controls frequency of the first channel waveform. It has the form "fx" where x is an integer greater than 0 or a constant can be used.

`frequency2 .....` the name of the optional parameter which controls frequency of the second channel waveform. It has the form "fx" where x is an integer greater than 0 or a constant can be used.

`tableSize ......` the name of the parameter which controls the number of values in each of the individual amplitude and phase tables (both should

have the same number of entries). It has the form "nx" where x is an integer greater than 0 or a constant can be supplied. Note that the total table size will be `2 x tableSize` because of the two channels.

`stepTime .....` the name of the parameter which controls the duration of each amplitude step. It has the form "dx" where x is an integer greater than 0 or a constant can be used.

Notes:

1. The pulse generated by this command has a length which includes a preferences defined delay called the pulse gate overhead (pgo). This delay is used to set up the RF gate – a signal sent to the high powered amplifier to switch on the DC biasing before the RF pulse is actually sent. You should account for this delay when writing the relationships list described above (the predefined variable "pgo" (pulse gate overhead) can be used. The minimum pgo possible with this command 5 μs.

3. There are a maximum number of amplitude steps available determined by the amount of DSP memory available. The total for all amplitude and phase tables must be less than 128 k values.

4. The stepTime value may range from 4.5 μs to 327,670 μs. **Beware**: values below 4.5 μs can cause the RF to switch on for very long times.

8. Potentially the shaped RF pulse could have a large amplitude and last for a long time – the combination of these situation could damage a high-powered RF amplifier or probe. Always take care when setting the amplitude and setTime values. Place limits where possible.

9. To update the shaped RF pulse between scans requires resending the tables to the DSP however the phaseOffset parameter can be applied using a phase cycle table to adjust the phase by a constant amount based on the scan number.

10. In addition to the pgo delay at the beginning of the command it takes 1 μs for the command to finish before another command can be given. Consequently you should always place a delay of at least 1 μs between rf pulse commands.

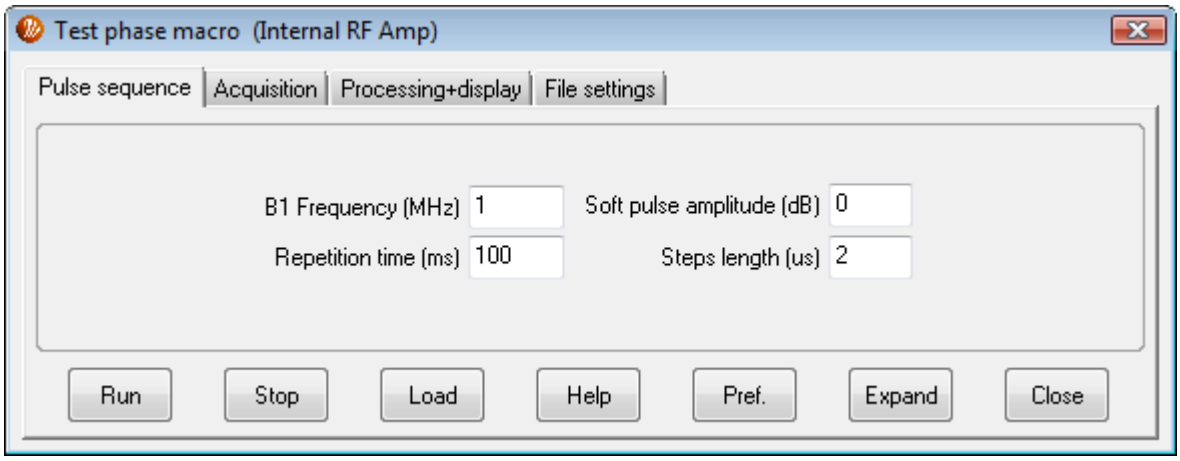
11. The total duration (in μs) of this command will be `pgo + tableSize*stepTime`.

Example pulse sequence:

```
############################################################
#
# A pulse sequence suitable for demonstrating the dual
# channel shaped pulse
```

```
#
############################################################

procedure(pulse_program,dir,mode,pars)

# Interface description (name, label, x, y, ctrl, vartype)
  interface = ["nucleus",       "Nucleus",                        "0","0", "tb",
"readonly_string",
               "b1Freq1",       "Shaped pulse freq ch1 (MHz)",    "0","1", "tbw",
"freq",
               "b1Freq2",       "Shaped pulse freq ch2 (MHz)",    "0","2", "tbw",
"freq",
               "repTime",       "Repetition time (ms)",           "0","3", "tbw",
"reptime",
               "90Amplitude1H", "Pulse amplitude (dB)",           "1","0", "tb",
"pulseamp",
               "pulseLength1H", "Pulse length (us)",              "1","1", "tb",
"pulselength",
               "ns",            "Nr. soft steps",                 "1","2", "tb",
"integer",
               "spAmp",         "Soft-pulse 1 amp(us)",           "1","3", "tb",
"pulseamp",
               "spDur",         "Soft-pulse step size(us)",       "1","4", "tb",
"float,[4.5,100]",
               "acqDelay",      "Pulse acqu. delay (us)",         "2","0", "tb",
"sdelay"]

# Relationships to determine remaining variable values
   relationships = ["nDataPnts = nrPnts",
                    "b1Freq1H  = b1Freq1",
                    "a90Amp    = 90Amplitude1H",
                    "d90Dur    = pulseLength1H",
                    "dAcqDelay = acqDelay",
                    "sa1       = sin(linvec(-1,1,ns)*2*pi+1e-8)/(linvec(-
1,1,ns)*2*pi+1e-8)",
                    "sa2       = linvec(0.1,1,ns)",
                    "t1        = ctor(sa1+i*sa2)",
                    "t1        = ucsRun:convertTxGain(spAmp)*mag(t1)",
                    "n1        = size(t1)/2",
                    "sp1       = 32768*(sa1 <= 0)",
                    "sp2       = 0*(sa2 >= 0)",
                    "t2        = ctor(sp1+i*sp2)",
                    "d1        = spDur",
                    "f1        = double(b1Freq1)",
                    "f2        = double(b1Freq2)",
                    "totPnts   = nrPnts",
                    "totTime   = acqTime"]

# Define the tabs and their order
   tabs = ["Pulse_sequence","Progress","Acquisition",
           "Processing_Std","Display_Std","File_Settings"]

# These parameters may be changed between experiments
   variables = [""]

# x and y spacing between controls
   dim = [170,26]

# Pulse sequence
   initpp(dir) # Reset internal parameter list
  # settxfreqs(f1,f2)
   delay(10)
   pulse(1,a90Amp,p1,f1,2,a90Amp,p1,f2,d90Dur)
   delay(10)
   shapedrf(t1,t2,p2,p3,f1,f2,n1,d1)
   delay(10)
   acquire("overwrite",nDataPnts)   # Acquire echo and wait

   lst = endpp() # Return parameter list

# Phase cycle list
  phaseList  = [0;   # Pulse phase
                0;   # ch1 table phase
                2;   # ch2 table phase
                0]   # Acquire phase

endproc(lst,tabs,interface,relationships,variables,dim,phaseList)
```

Test parameters



Output, showing shaped RF pulse (blue) and TTL gate pulse (green) with pgo = 2 μs.