

DRV2624 FREQUENTLY ASKED QUESTIONS

The DRV2624 is a versatile Haptics driver for ERM & LRA motors that can operate in both open and closed loop operation. This device has 8 different modes that can be optimized for most Haptics applications. In order to play a Haptics waveform, the user must program the device's built in driver appropriately to get optimized performance. Internally, the DRV2624 contains a 1K byte RAM that can be used to store waveforms that will be repeated during operation, or the user can play directly via the Real-Time-Playback port (RTP) or directly via a PWM input.

Table of Contents

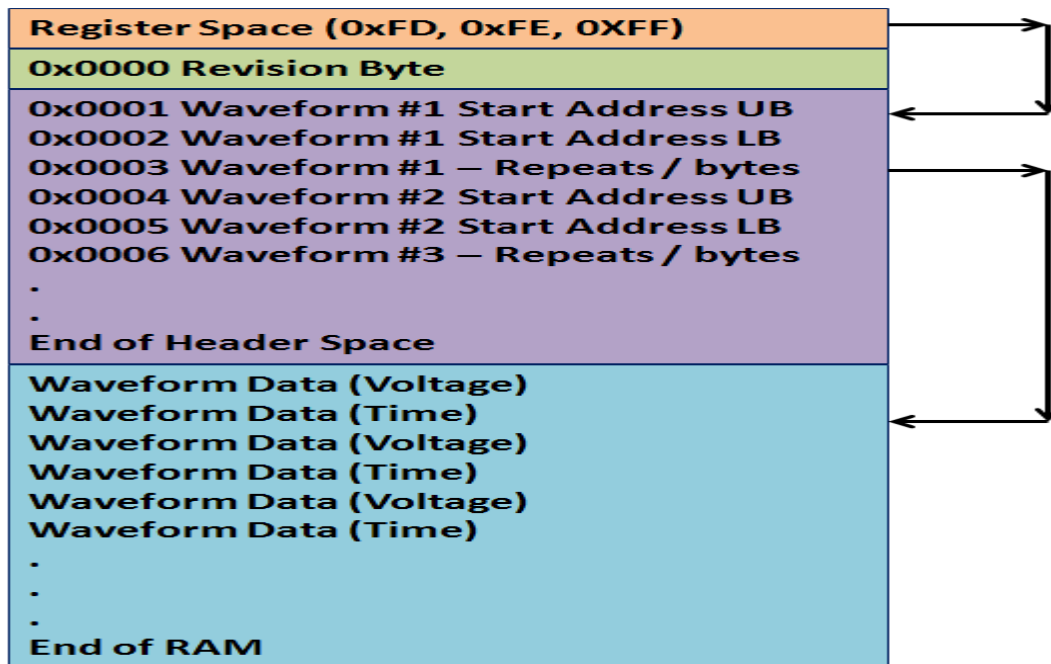
| | |
|---|---|
| How can I Play a Waveform from RAM? | 2 |
| How can I play a “buzz” from RAM?..... | 2 |
| Multi-byte Writes via I2C..... | 4 |
| How can I play a “ramp-up” Waveform from RAM? | 4 |
| How can I play a “ramp-down” Waveform from RAM?..... | 6 |
| How can I combine waveforms to produce a more complicated Haptics effect? | 8 |
| How can insert wait periods into a Haptics effect?..... | 8 |

HOW CAN I PLAY A WAVEFORM FROM RAM?

In order to play a waveform from RAM, the user must program both the Waveform Sequence and the RAM. Once the Waveform sequence and RAM are programmed correctly, playing a waveform is performed by a single I2C write to the “GO” bit or trigger from the TRIG/INTZ pin. Once the RAM is programmed, the Waveform Sequences in register space can be used to mix and match waveform effects to produce the desired motor actuation.

Although the internal RAM needs to be reinitialized after every power-up cycle, once programmed, the RAM retains its values until power is removed. For ease of use, the user should simply program the RAM once at power-up and reference the stored waveforms via their waveform number. Since the only way to access the configuration space and RAM is through the I2C interface. There are pointers stored in the configuration space that are used to point to the location of RAM where the waveforms are stored.

The RAM is divided into 3 sections starting with the Revision Byte stored at 0x0000. The next section of the RAM is the Header Space starting at location 0x0001. The Header space is a variable length that will contain the starting address and number of bytes and a repeat count for each waveform the used intends to use. Finally the various waveform voltage & time pairs should be programmed in the remainder of the available RAM space.



How can I play a “buzz” from RAM?

The following example will program a single “buzz” waveform into the RAM followed by playing the waveform. This single waveform example is just done in order for the reader to understand the steps in programming a waveform sequence and thus repeat these steps are necessary to fully program the device for all the desired waveforms.

Let’s set up the “buzz” waveform that has the following parameters:

Waveform value: Play a constant 25% max amplitude waveform for 200 ms

Voltage value is 0x10 which represents 25% of 127 (BiDir_Input = 0; Open_Loop = 1).

Time value is 200ms/5ms = 40 ticks or (0x28 time).

Waveform number: 0x07

In this example the value of 0x07 was chosen at random. Further for this example, it is assumed that waveforms 1-6 are already programmed to some other values that are will not be covered in this example. However, the value of “0x07” indicates the following:

The Waveform Upper byte is located at 0x13 in the RAM Header space

The Waveform Lower byte is located at 0x14 in the RAM Header space

The number of bytes & repeats are located at 0x15 in the RAM Header space

Each waveform number has 3 bytes of informing stored in a **fixed location in the RAM header space** starting with address 0x0001-0x0003 for waveform #1, 0x0004-0x0006 for waveform #2 etc. The amount of waveforms that can be stored is a function of the number of bytes it takes to define each waveform effect.

Location in RAM Header Space:

0x13 - Value 0x00 for Upper Address Byte

0x14 - Value 0x64 for Lower Address Byte

0x15 - Value 0x02 to indicate 0 Repeats and 2 Bytes of data (1 voltage/time Pair)

It takes 3 I2C writes to program each byte of RAM. The Upper and Lower Address bytes as well as the RAM Data the user wishes to write all need to be defined. The first step is to program the RAM is to program the Header Space with the following I2C writes:

| I2C Address | I2C Data | Description |
|-------------|----------|--|
| 0x01 | 0x01 | Take the device out of STANDBY & set internal trigger mode |
| 0x04 | 0x07 | Set the 1 st waveform sequence to Waveform #7 |
| 0x05 | 0x00 | Set the Terminate Character - Optional but good practice since the default is 0x00 |
| 0xFD | 0x00 | Set the Upper 8 bits of RAM Address |
| 0xFE | 0x00 | Set the Lower 8 bits of RAM Address |
| 0xFF | 0x01 | Set the Revision byte (not used by the device) in RAM - The revision byte is the 0x0000 location in RAM that the user can access to determine what version of the library and effects are loaded into the DRV2604. |
| 0xFE | 0x13 | Set the Lower 8 bits of RAM Address |
| 0xFF | 0x00 | Upper byte waveform Starting Address in the RAM at location 0x0013 |
| 0xFF | 0x64 | Lower byte waveform Starting Address in the RAM at location 0x0014 |
| 0xFF | 0x02 | 0 Repeats and 2 Bytes of data in the RAM at location 0x0015 |
| 0xFE | 0x64 | Set the Lower 8 bits of RAM Address |
| 0xFF | 0x10 | Voltage Value |
| 0xFF | 0x28 | Time Value |
| 0x0C | 0x01 | Set the "GO" bit - This will start the waveform |

To take advantage of the **auto increment RAM Address** feature in the device, the RAM Address registers do not need to be re-written for each byte written to the RAM. Notice in step “f” above, the Lower RAM Address (0xFE) is reset to 0x13, followed by 3 consecutive writes to 0xFF (steps g-i). In reality locations 0x0013, 0x0014 & 0x0015 are being written inside the RAM. The same sort of feature is also utilized on step “j” above.

At this point, the contents of RAM will contain the following:

| DRV2604 RAM File Structure Example | | |
|---|---------------------------|-------------|
| Address | Description | Data |
| 0x0000 | Revision Byte | 0x00 |
| 0x0001 | Waveform 1 Upper Byte | TBD |
| 0x0002 | Waveform 1 Lower Byte | TBD |
| 0x0003 | Waveform 1 Repeat / bytes | TBD |
| | ... | |
| 0x0013 | Waveform 7 Upper Byte | 0x00 |
| 0x0014 | Waveform 7 Lower Byte | 0x64 |
| 0x0015 | Waveform 7 Repeat / bytes | 0x02 |
| | ... | |
| 0x0064 | Waveform 7 Voltage 1 | 0x10 |
| 0x0065 | Waveform 7 Time 1 | 0x28 |

Multi-byte Writes via I2C

Since the DRV2624 comprehends multi-byte I2C writes, the above transaction can be further simplified. Notice in the above example, there are multiple writes to registers 0xFF. In order to simplify this transaction, the user can further take advantage of the DRV2624's built in auto-increment feature when accessing the RAMData byte (0xFF). By using multi-byte I2C transactions, the set-up and play procedure is greatly simplified (from 13 I2C transactions to 6 transactions) as follows:

Write 0x00 to register 0x01 – take the device out of its low power state & set internal trigger mode

Write 0x07 to register 0x04 – set the 1st waveform sequence to Waveform 7

Since register 0x05 defaults to “00”, there is no need to set the “terminate character”

Using 3 multi-byte I2C transactions, write the following:

Starting at 0xFD, write 0x00, 0x00, 0x00 – sets the Revision byte (not used by the device)

Starting at 0xFD, write 0x00, 0x13, 0x00, 0x64, 0x02 – Programs the Header information (Starting Address (0x0064), Number of Repeats & Number of Bytes (0x02))

Every time the I2C accesses register space location 0xFF, it automatically increments register 0xFD & 0xFE in auto increment the RAM address.

Starting at 0xFD, write 0x00, 0x64, 0x10, 0x28 – sets the Voltage/time pair

Write 0x01 to register 0x0C – This will start the waveform

How can I play a “ramp-up” Waveform from RAM?

The following example will program a single “ramp-up” waveform into the RAM followed by playing the waveform.

An example of setting up a “ramp-up” waveform that has the following parameters:

Waveform value: Start from “0” and ramp the amplitude to full-scale in 100 ms

Voltage value starts at 0x00 and ends at 0x3F (BiDir_Input = 0; Open_Loop = 1).

Time value is 100ms/5ms = 20 ticks or (0x14 time).

Waveform number: 0x08. Since waveform 7 was previously defined, and can be re-used later, choosing a different waveform adds to our suite of pre-defined effects.

The value of “0x08” indicates the following:

The Waveform Upper byte is located at 0x16 in the RAM Header space

The Waveform Lower byte is located at 0x17 in the RAM Header space

The number of bytes & repeats are located at 0x18 in the RAM Header space

Location in RAM Header Space:

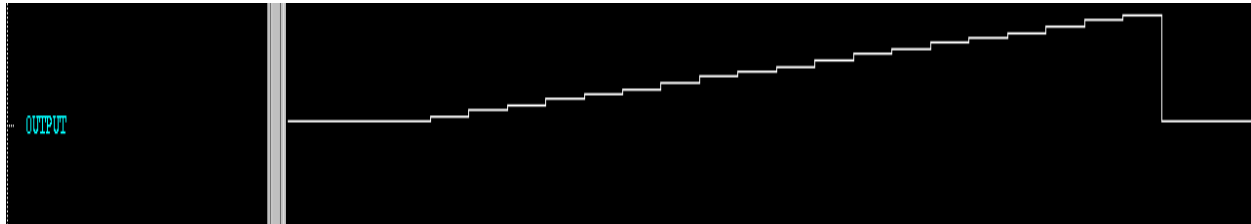
0x16 - Value 0x00 for Upper Address Byte

0x17 - Value 0x66 for Lower Address Byte

0x18 - Value 0x04 to indicate 0 Repeats and 4 Bytes of data (2 voltage/time pairs)

These are the steps to program the waveform:

| I2C Address | I2C Data | Description |
|-------------|------------------------------------|---|
| 0x01 | 0x00 | Take the device out of its STANDBY & set INTERNAL TRIGGER mode. |
| 0x04 | 0x08 | Set the 1st waveform sequence to Waveform 8 |
| 0x05 | 0x00 | Set TERMINATE CHARACTER; this will ensure only waveform 8 plays. |
| 0xFD | 0x00, 0x00, 0x00 | Using a multi-byte I2C write transaction; Set the Revision byte (not used by the device). |
| 0xFD | 0x00, 0x16, 0x00, 0x66, 0x04 | Using a multi-byte I2C write transaction; Programs the Header information (Starting Address, Number of Repeats & Number of Bytes) |
| 0xFD | 0x00, 0x66, 0x80, 0x14, 0x3F, 0x00 | Using a multi-byte I2C write transaction; Set the Voltage/time pairs; Notice that the MSB of the 1st Voltage/Time pair is set indicating the part should ramp from the 1st Voltage to the 2nd Voltage. Also notice the time of the ramp is set on the 1st Voltage, not the 2nd Voltage. |
| 0x0C | 0x01 | Set the "GO" bit to start the waveform |



Notice in the waveform above, there are 20 step sizes ramping the waveform from 0x00 to 0x3F

After adding a 2nd waveform to the RAM, the contents will now contain the following:

| DRV2604 RAM File Structure Example | | |
|---|---------------------------|-------------|
| Address | Description | Data |
| 0x0000 | Revision Byte | 0x00 |
| 0x0001 | Waveform 1 Upper Byte | TBD |
| 0x0002 | Waveform 1 Lower Byte | TBD |
| 0x0003 | Waveform 1 Repeat / bytes | TBD |
| | ... | |
| 0x0013 | Waveform 7 Upper Byte | 0x00 |
| 0x0014 | Waveform 7 Lower Byte | 0x64 |
| 0x0015 | Waveform 7 Repeat / bytes | 0x02 |
| 0x0016 | Waveform 8 Upper Byte | 0x00 |
| 0x0017 | Waveform 8 Lower Byte | 0x66 |
| 0x0018 | Waveform 8 Repeat / bytes | 0x04 |
| | ... | |
| 0x0064 | Waveform 7 Voltage 1 | 0x10 |
| 0x0065 | Waveform 7 Time 1 | 0x28 |
| 0x0066 | Waveform 8 Voltage 1 | 0x80 |
| 0x0067 | Waveform 8 Time 1 | 0x14 |
| 0x0068 | Waveform 8 Voltage 2 | 0x3F |
| 0x0069 | Waveform 8 Time 2 | 0x00 |

How can I play a “ramp-down” Waveform from RAM?

The following example will program a single “ramp-up” waveform into the RAM followed by playing the waveform.

Let’s set up the “ramp-down” waveform that has the following parameters:

Waveform value: Start from “3F” and ramp the amplitude to mid-scale in 100 ms

Voltage value starts at 0x3f and ends at 0x00 (BiDir_Input = 0; Open_Loop = 1).

Time value is 100ms/5ms = 20 ticks or (0x14 time).

Waveform number: 0x09

The value of “0x09” indicates the following:

The Waveform Upper byte is located at 0x19 in the RAM Header space

The Waveform Lower byte is located at 0x1A in the RAM Header space

The number of bytes & repeats are located at 0x1B in the RAM Header space

Location in RAM Header Space:

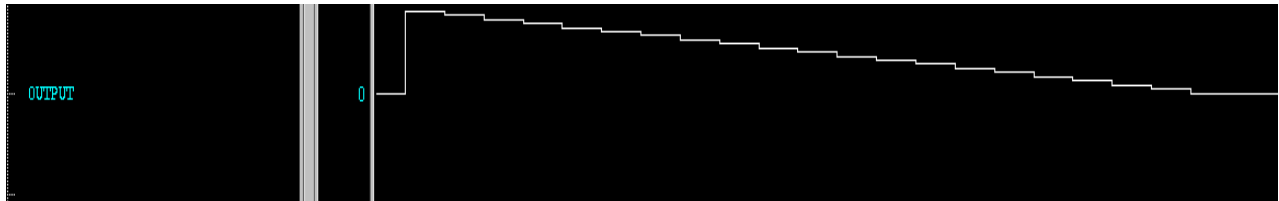
0x19 - Value 0x00 for Upper Address Byte

0x1A - Value 0x6A for Lower Address Byte

0x1B - Value 0x04 to indicate 0 Repeats and 4 Bytes of data (2 voltage/time pairs)

The exact I2C transactions for this waveform would be as follows:

| I2C Address | I2C Data | Description |
|-------------|------------------------------------|--|
| 0x01 | 0x00 | Take the device out of its STANDBY & set INTERNAL TRIGGER mode. |
| 0x04 | 0x09 | Set the 1st waveform sequence to Waveform 9 |
| 0x05 | 0x00 | Set TERMINATE CHARACTER; this will ensure only waveform 9 plays. |
| 0xFD | 0x00, 0x00, 0x00 | Using a multi-byte I2C write transaction; Set the Revision byte (not used by the device). |
| 0xFD | 0x00, 0x19, 0x00, 0x6A, 0x04 | Using a multi-byte I2C write transaction; Programs the Header information (Starting Address, Number of Repeats & Number of Bytes) |
| 0xFD | 0x00, 0x6A, 0xBF, 0x14, 0x00, 0x00 | Using a multi-byte I2C write transaction; Sets the Voltage/time pairs; Notice that the MSB of the 1st Voltage/Time pair is set indicating the part should ramp from the 1st Voltage to the 2nd Voltage. Also notice the time of the ramp is set on the 1st Voltage, not the 2nd Voltage. |
| 0x0C | 0x01 | Set the "GO" bit to start the waveform |



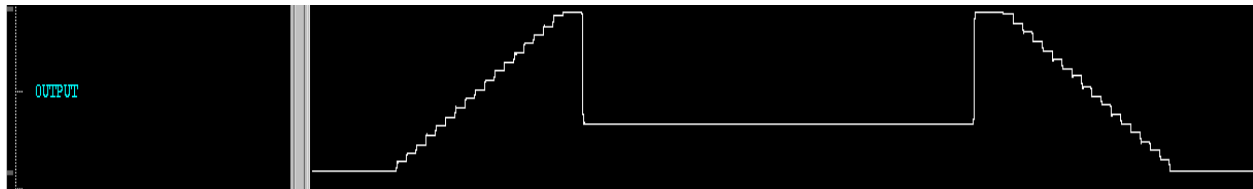
Similar to the ramp-up waveform, the ramp-down also takes 20 5-ms steps per the programmed instructions.

After all three waveforms are loaded in the RAM, the contents will now contain the following:

| DRV2604 RAM File Structure Example | | | | | | |
|------------------------------------|---------------------------|------|--|---------|----------------------|------|
| Address | Description | Data | | Address | Description | Data |
| 0x0000 | Revision Byte | 0x00 | | 0x0064 | Waveform 7 Voltage 1 | 0x10 |
| 0x0001 | Waveform 1 Upper Byte | TBD | | 0x0065 | Waveform 7 Time 1 | 0x28 |
| 0x0002 | Waveform 1 Lower Byte | TBD | | 0x0066 | Waveform 8 Voltage 1 | 0x80 |
| 0x0003 | Waveform 1 Repeat / bytes | TBD | | 0x0067 | Waveform 8 Time 1 | 0x14 |
| | ... | | | 0x0068 | Waveform 8 Voltage 2 | 0x3F |
| 0x0013 | Waveform 7 Upper Byte | 0x00 | | 0x0069 | Waveform 8 Time 2 | 0x00 |
| 0x0014 | Waveform 7 Lower Byte | 0x64 | | 0x006A | Waveform 9 Voltage 1 | 0xBF |
| 0x0015 | Waveform 7 Repeat / bytes | 0x02 | | 0x006B | Waveform 9 Time 1 | 0x14 |
| 0x0016 | Waveform 8 Upper Byte | 0x00 | | 0x006C | Waveform 9 Voltage 2 | 0x00 |
| 0x0017 | Waveform 8 Lower Byte | 0x66 | | 0x006D | Waveform 9 Time 2 | 0x00 |
| 0x0018 | Waveform 8 Repeat / bytes | 0x04 | | | | |
| 0x0019 | Waveform 9 Upper Byte | 0x00 | | | | |
| 0x001A | Waveform 9 Lower Byte | 0x6A | | | | |
| 0x001B | Waveform 9 Repeat / bytes | 0x04 | | | | |
| | ... | | | | | |

How can I combine waveforms to produce a more complicated Haptics effect?

It is possible to combine the 3 above examples into a single transaction using the waveform sequencer. To create a waveform that would ramp-up, buzz then ram-down, simply program Waveform Sequencer 1, 2 & 3 registers to play the ramp-up (waveform 8 in register 0x04), followed by the buzz (waveform 7 in register 0x05), followed by the ramp-down (waveform 9 in register 0x06). Below is the output by combining the 3 previous examples into a single playback. Since this “combined waveform” is much longer than each of the single waveforms, the time-scale might look a little skewed.

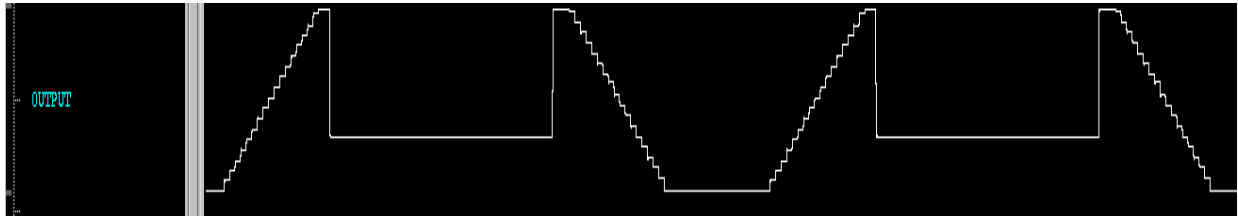


As you can see, creating custom waveforms can be easily programmed. Further, by combining the custom waveforms, the DRV2624 can be used to produce some very complicated Haptics effects.

How can insert wait periods into a Haptics effect?

The “wait” feature can also be used in conjunction with programmed Haptics waveforms. To build upon this example even further, we can utilize the “wait” feature by programming the following:

- a) Waveform Sequencer 1 (0x04) – set to value 0x08 (ramp-up waveform)
- b) Waveform Sequencer 2 (0x05) – set to value 0x07 (buzz waveform)
- c) Waveform Sequencer 3 (0x06) – set to value 0x09 (ramp-down waveform)
- d) Waveform Sequencer 4 (0x07) – set to value 0x89 (play a wait for 90 ms)
- e) Waveform Sequencer 5 (0x08) – set to value 0x08 (ramp-up waveform)
- f) Waveform Sequencer 6 (0x09) – set to value 0x07 (ramp-up waveform)
- g) Waveform Sequencer 7 (0x0A) – set to value 0x09 (ramp-up waveform)
- h) Waveform Sequencer 8 (0x0B) – set to value 0x00 (terminate character)
- i) Write 0x01 to register 0x0C – Set the “GO” bit to start the waveform



Notice in step (d) above, Sequence 4 was programmed to 0x89. When the MSB of one of the Waveform Sequence registers is set to a “1” it will produce a “wait” of 10ms * the value instead of playing a stored waveform. So in this example, a 0x09 will play the waveform stored in the RAM (as in steps (c) & (g)), and a 0x89 will produce a 90ms wait pattern that can be inserted before or after another waveform.