

APPLICATIONS BRIEF 42 — IMPLEMENTING A PAUSE FUNCTION IN THE ISD4000 DEVICE SERIES

One feature usually desired in pocket recorders is the ability to pause a record operation and then to restart the record seamlessly at the point recording was paused. This function may be implemented into an ISD4000 Series device using a very simple software algorithm. The discussion below explains how this may be accomplished.

DISCUSSION

The ISD4000 device internal hardware automates the record function. Anytime a record operation is ended by a STOP command, an EOM bit is written into the device to mark that location. This works acceptably for a linear message record operation. A PAUSE during record function, however, requires the chip stop recording then restart later without an EOM being present. This type of operation is not built into the ISD4000. Fortunately, this function is easy to implement in software.

The steps to implement a PAUSE RECORD function follow:

1. A normal RECORD operation is begun and the user records for a time then wished to pause.
2. The user pauses record by pressing a "PAUSE" button built into the record system. The PAUSE button is connected to the microcontroller and causes the PAUSE RECORD algorithm to be executed.
3. The PAUSE RECORD algorithm first allows the record operation to continue without stopping for the length of one row inside the ISD4000 device. For example, if an ISD4003-4M device is being used, each row is 200 milliseconds long (see Application Brief #38, "The ISD ChipCORDER Product Line: Sizes, Row Lengths and Other Useful Details for Addressable ISD products"). This insures that the EOM will be written in a row that contains sound recorded after the PAUSE button was pressed.

4. After the STOP command has been executed, the microcontroller now issues a dummy command with the RUN bit set LOW to read the current address location from the ISD4000. This address that is now read back will be for the row following the row with EOM marker that was just recorded. The microcontroller then subtracts one from this number and stores it in a temporary register.
5. The user now presses the PAUSE button again to continue recording. The microcontroller retrieves the address from the temporary register and then loads it into the ISD4000 and starts a RECORD operation. This erases the EOM and starts recording the additional information in the row that previously contained the EOM. Since this row contained sounds recorded after the PAUSE button was pressed, none of the desired recording was lost.

When this message is played back, the message will be played in total without stopping at the point where the PAUSE button is pressed.

It is also possible to implement a PAUSE PLAYBACK function:

1. A normal PLAY operation is begun to allow the user to listen to a previously recorded message. They then wish to "pause" the playback.
2. The user pauses playback by pressing a "PAUSE" button built into the record system. The PAUSE button is connected to the microcontroller and causes the PAUSE PLAYBACK algorithm to be executed.
3. The PAUSE PLAYBACK algorithm immediately issues a STOP command to the ISD4000 device.

4. After the STOP command has been executed, the microcontroller again issues a dummy command with the RUN bit set LOW to read the current address location from the ISD4000. This address that is now read back will be for the row after the row where playback was halted. The microcontroller then subtracts one from this number and stores it in a temporary register.
5. The user now presses the PAUSE button again to continue playback. The microcontroller loads the address from the temporary register into the ISD4000 and issues a playback command. Playback begins at the beginning of the row and continues for the rest of the message.

The two simple algorithms above have demonstrated how a PAUSE function may be implemented for both record and playback operations in the ISD4000 series device. In each case, only a small amount of assembly or C language code is required to implement the function.