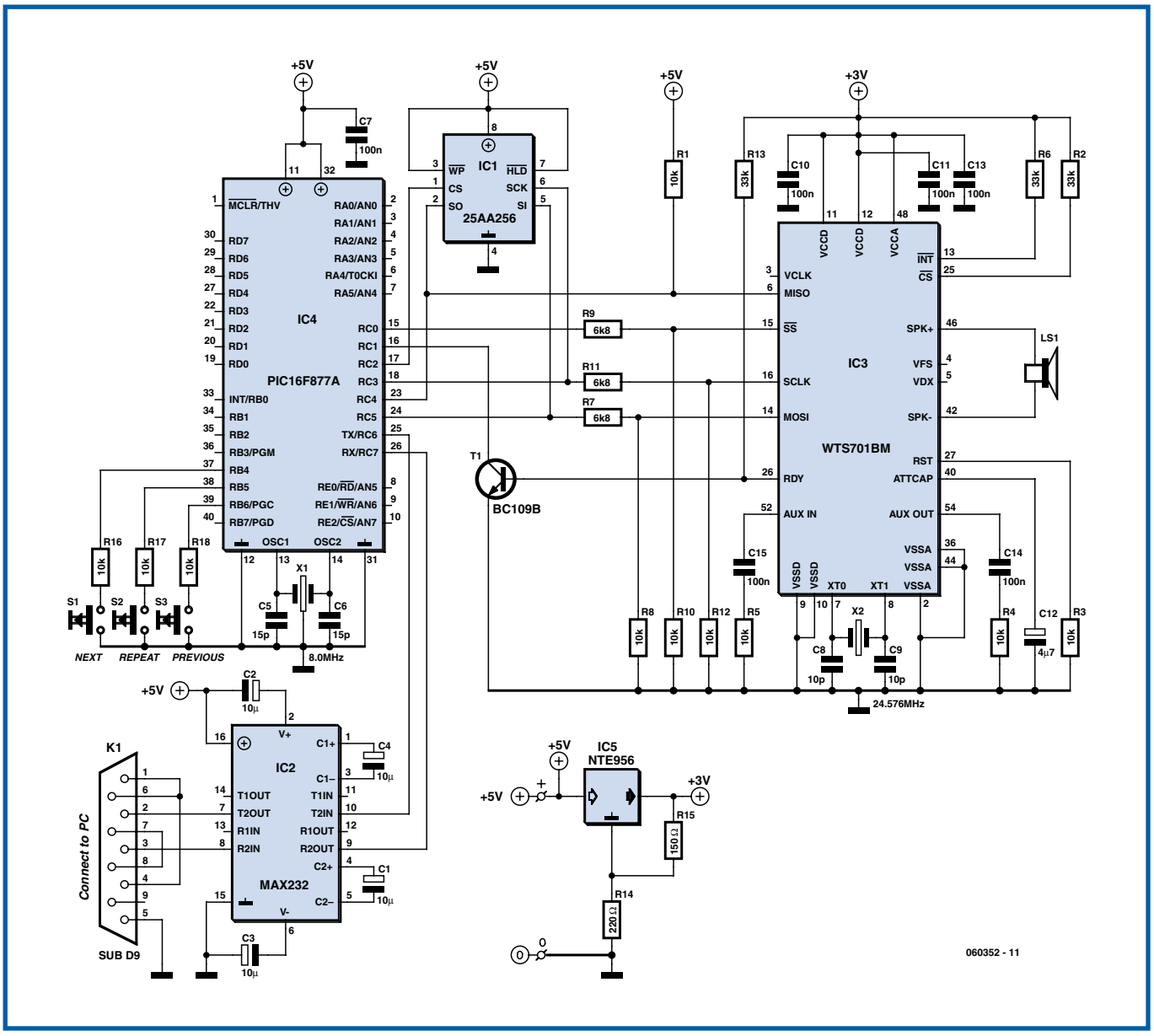
E-Dictator



Alexander Pozhitkov

Here is a microcontroller-based system that acts like a speech-enabled reminder with a very intuitive interface.
The user interface of the E-dictator is very intuitive and simple. Upon turning the system on, it is immediately able to reproduce stored messages or record new ones. All messages recorded into E-dictator are stored in non-volatile memory.

To record new messages, E-dictator must be connected to a PC via a serial port and connector K1. Any terminal program (such as HyperTerminal) set to 19200, 8, N, 1 should do. No commands whatsoever are necessary to make E-dictator start recording: just paste your text into the message window or use Send Text File from the Transfer menu (HyperTerminal). Every character received by E-dictator is echoed to the PC allowing the user to visually check the data transfer, at then end of which E-dictator automatically stops receiving.

Paragraphs of uploaded text are separated by a carriage return (CR). There is no limit to the size of the paragraphs (i.e. message). The only limit is the total volume of text that can be uploaded, which is determined by the capacity of EEPROM (IC1). Currently, 32 kbits of text can be uploaded into E-dictator.

Once the text has been uploaded, the buttons 'Next', 'Repeat' and 'Previous' will prompt E-dictator to reproduce the messages.
The speech processor and EEPROM interact with the microcontroller via an SPI interface, which is implemented via RC3, RC4 and RC5 pins of the micro. The ICs are activated one at the time via RC0 and RC2 outputs of the microcontroller. Voice output is by way of a miniature 8-Ω loudspeaker.

The speech processor (IC3) is a Winbond device is produced in four versions: male or female voices for English or Chinese languages. Its processor has SPI connectivity covering data, control and status. Of the other pins that can be used to monitor the status of

the processor, E-dictator employs only one, called 'Ready' (RDY). Resistors R7-R12 act as voltage dividers to adjust 5-V logic to 3-V logic on the speech processor input lines. The open-drain output of the speech processor (MISO) is tolerant of 5-V swings, allowing a seamless connection of the speech processor to the microcontroller. RDY is an open-drain feedback pin pulling Low when the internal buffer of the speech processor is full. RDY was found to be 5-V intolerant, i.e. major faults would occur in the operation of IC3 if this pin was pulled up to 5 V. Therefore, transistor T1 is necessary to adjust voltage levels.

The software was written in assembly language using Microchip IDE. The PIC program is organized in an event-driven fashion. An event loop) is constantly running upon the program start. Two conditions cause the loop to break: occurrence of a data character in the UART buffer or a press-button event.

Recording messages. When data arrive to the UART buffer, the program switches to a message recording mode. In this mode, as soon as the character arrives to the UART buffer it is echoed back to the PC and also immediately transferred into EEPROM via the SPI. The EEPROM can only continuously receive up to 64 bytes, which is determined by its paged organization. As soon as the 64-byte page is full, the EEPROM needs some time (up to 5 ms) to actually store the data. While the EEPROM is completing its internal write cycle, the PC does not stop transmitting characters. Those characters are temporarily stored in the RAM of the microcontroller until the EEPROM is ready for more data. As soon as that is the case, the stored characters are flushed into the EEPROM. After that, the microcontroller returns to its regular procedure of direct transmission of the data from the UART to the EEPROM. Recording mode stops when a timeout occurs in the data transmission, which causes the program flow to return to the event loop.

Playing back messages. When you press a button, this event is registered and the program flow is diverted from the event loop to a speaking mode. The PIC micro maintains a pointer to the current message. If the user pressed 'Repeat', then microcontroller retrieves text from the EEPROM starting from the appropriate address. Due to the EEPROM and speech processor sharing the same SPI bus, the PIC micro stores a portion of the text message in its internal RAM and uploads text into the speech processor piece by piece: while the speech processor is interpreting and speaking the text, the microcontroller retrieves further parts of the message from the EEPROM and stores in its RAM. This procedure continues until the entire message is transferred from the EEPROM to the speech processor.

If the user presses 'Previous' or 'Next', the program scrolls through EEPROM to locate the appropriate message. When it is located, the program executes the same procedure as described in the previous paragraph.

(060352-I)

The project software is available as a free download from www.elektor.com. Follow Magazine → October 2007 → Mailbox.

Advertisement