EE389 Electronics Design Lab II, IIT Bombay, November 2008

Hospital Alarm Clock

Group No: D6

Saurabh Dubey (05D07004) sorvajdb@ee.iitb.ac.in

Rahul Jain (05D07032) rahulj@ee.iitb.ac.in

Avishek Majumdar (05D07038) avishek@ee.itb.ac.in

Supervisor – Abhay Karandikar

Abstract

The following report implements a Hospital Alarm Clock which play pre-recorded medicine names stored by the user at the fed times. Atmega8535 is being used which stores and also play backs the data in DataFlash AT90S8535 using SPI interface. The voice signal is being sampled at the rate of 8 kHz with each sample of 8 bits. The output is being played using 8-bit PWM followed by RC delay and amplifier.

1. Introduction

In a hospital, there are patients with each patient having requiring different medicines at different time intervals and having different dosages. This is makes it very tough to keep track whether he has taken the medicines are not. This creates a demand of a device which reminds a user that this medicine needs to be given to a patient at the particular time. We have tried to solve the same problem here by providing a novel product which plays out which all medicines needs to be taken out by the patient at the proper time.

2. Implementation

Initially, the ports have to be set up. This is done in the “setup” subroutine. The SPI protocol defines one device as a master and the other devices connected to this master as slaves. In this application the AVR microcontroller functions as a master and the DataFlash as a slave. As the AT90S8535 is the only master in this application the SS pin can be used as an I/O pin.
The SPI of the AT90S8535 is defined as an alternative function of Port B (PB5 to PB7). In this application the control signals for the DataFlash are also set up on Port B (PB0 to PB2 and PB4). The free pin (PB3) is used to control the status LED.

Then, PortB is set to a defined status with all outputs high and internal pull-up resistors on the inputs. The A/D converter of the AT90S8535 is connected to PortA. Therefore, PortA is defined as a high-impedance input.

PortD serves as an input for the pushbuttons and as an output for the PWM signal. Here the PWM function of Timer1 on the output pin PD4 is used.

In the end, interrupts are enabled. In this application two interrupts (“ADC” and “Timer1 Overflow”) are used, which are enabled and disabled directly in the subroutine when they are required.

2.1 Main Loop

In the main loop, the three pushbuttons are scanned. If one of them is pressed, the LED is turned on to show that the system is busy and the corresponding subroutine is called. An extra loop is performed, until the button is released, as a software debounce for the “Erase” and “Playback” functions.

During the main loop, the LED is turned off to indicate that the system is running idle.

2.2 Erase Loop

When erase subroutine is called, a flag is set which indicates that in the next recording cycle the new data can be stored at the beginning of the DataFlash. The SPI has to be set up for accessing the DataFlash. No interrupts are used here. The data order for the DataFlash is MSB first and the AT90S8535 is the master.

The DataFlash accepts either the SCK signal being low when CS toggles from high to low (SPI Mode 0) or the SCK signal being high when CS toggles from high to low (SPI Mode 3) with a positive clock phase. In this application the SPI is set up in Mode 3. In order to get the fastest data transfer possible, the lowest clock division is chosen, running the SPI bus at 2 MHz using an oscillator crystal of 8 MHz.

To perform a block erase, the CS line is driven low and the opcode 0x50 is loaded into the DataFlash followed by two reserved bits (zeros), the 9-bit block address, and 13 don’t care bits. This sequence is transferred to the slave bytewise. After each byte, the SPI Status Register – SPSR – is checked until the SPI Interrupt Flag indicates that the serial transfer is complete. After the whole sequence is written, erasing of the block is started when the CS line is driven high.

The Ready/Busy pin is driven low by the DataFlash until the block is erased. Then the next block will be erased in the same way as the current. This takes place until all 512 blocks are erased. An erased location reads 0xFF.

2.3 Record

The record subroutine consists of the setup of the A/D converter and an empty loop which is performed as long as the “Record” button is pressed. The ADC0 pin is used in this application which requires the ADC Multiplexer Select Register (ADMUX) being set to zero. In the ADC Control and Status Register (ADCSR) the ADC is enabled with a clock division factor of 32, set to single conversion mode, interrupts enabled, and the interrupt flag is cleared. The A/D conversion is also immediately started. The first

conversion takes longer than the following conversions (832 oscillator cycles instead of 448). After this time, the ADC interrupt occurs indicating that the conversion is finished and the result can be read out of the ADC Data Register.
The analog signal from the microphone circuit is sampled at 8 kHz. To achieve a sampling frequency of 8 kHz, a sample has to be taken every 1000 cycles (8 kHz x 1000 = 8 MHz). To get one A/D conversion result, each 1000 clock cycles the ADC is run in single conversion mode with an ADC clock division by 32. A single conversion takes 14 ADC cycles. Therefore a conversion will be ready after 14 x 32 = 448 cycles.
When a conversion is finished an interrupt occurs. The interrupt routine then performs a loop to fill in the missing 100 - 448 = 552 cycles, before a new A/D conversion is started.

2.4 Write to Data Flash

Writing data to the DataFlash is done by writing first to a buffer and when this buffer is full writing it's contents to one page of the main memory.
To write data to the buffer, the CS line is driven low and the opcode 0x84 is loaded into the DataFlash. This is followed by 14 don't care bits and the 10-bit address for the position within the buffer. Then the 8-bit data is entered.
This sequence is transferred to the slave bytewise. After each byte the SPI Status Register – SPSR – is checked until the SPI Interrupt Flag indicates that the serial transfer is complete. After the whole sequence is written the CS line is driven high.
If the buffer is full and there are empty pages left, the buffer is copied to the next page of the DataFlash. As the memory has been erased earlier, data can be written without additional erasing.
If the memory is filled, a loop is executed until the "Record" button is released. Any data recorded while the memory is full will be lost.

2.5 Playback

In the "Playback" subroutine, the contents of the DataFlash are read out and modulated as an 8-bit PWM running at 15,686 Hz. For the PWM, the 16-bit Timer/Counter1 is used with the PWM output on OC1B. This is defined in the Timer/Counter Control Registers A and B (TCCRA/TCCRB). For running the PWM at the highest possible frequency, the PWM clock divider is set to 1.
When the set-up is done, the first page is copied into Buffer 1 by driving the CS line low and transferring the appropriate commands to the DataFlash. The page-to-buffer transfer is started when the CS line is driven high again. When the Ready/Busy pin is driven high by the DataFlash, Buffer 1 contains valid data. Then the next page transfer to Buffer 2 is started. As both buffers are independent from each other, data can already be read from Buffer 1 while the DataFlash is still busy copying data from the second page to Buffer 2.

For reading a byte from a buffer, a dummy value has to be written to the DataFlash. A write action of the master to an SPI slave causes their SPI Data Register – SPDR – to be interchanged. After writing a dummy byte to the DataFlash, the SPDR of the AVR microcontroller contains the output data from the DataFlash.

3. Possible Extensions in Future

- Increasing the memory for the number of messages stored and time for a pre-recorded voices are natural extensions.
- PC Interface can be provided which will download the calendar (e.g. Google Calendar). It can automatically read the calendar and then, it can play the proper medicines' name at the correct time.
- There can be an automatic jack which can come out at the time the pre-recorded voices are being played with the medicines.
- Here, 8-bit PWM is used. This is very error-prone and the reconstruction of the output is not very good in this case. So, 10-bit PWM can be used in future applications.
- PWM increases the analog circuitry and imput and output voltages need to be matched properly. So, instead of PWM, DAC can be used in future applications.

References

- Avr335: Digital Sound Recorder with AVR and DataFlash Application Note
- Application Note on implementation of Real Time Clock (RTC) on FPSLIC embedded AVR microcontroller

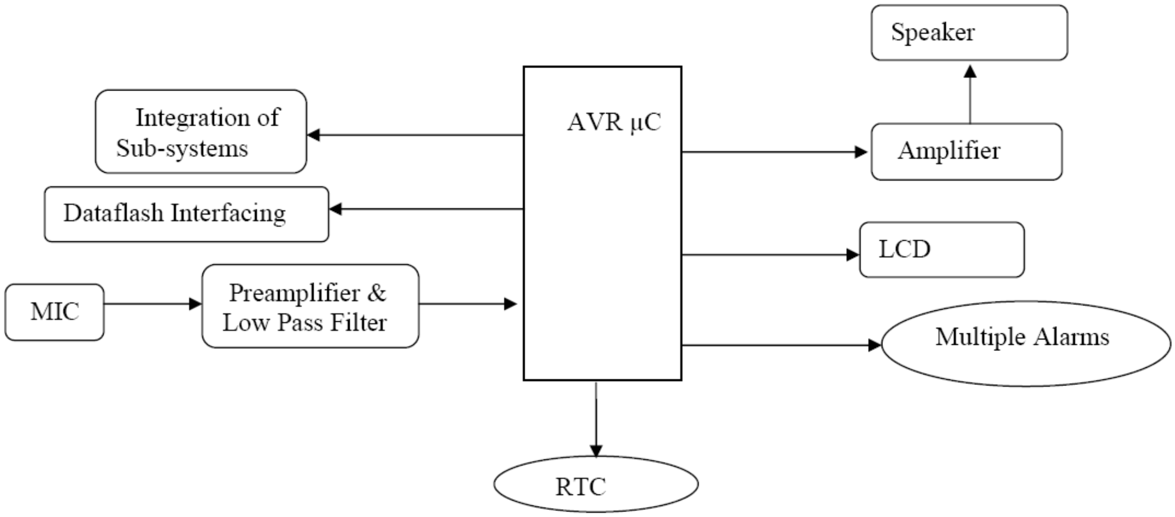

Fig. 1 - Block Diagram of the various tasks to be performed in the project

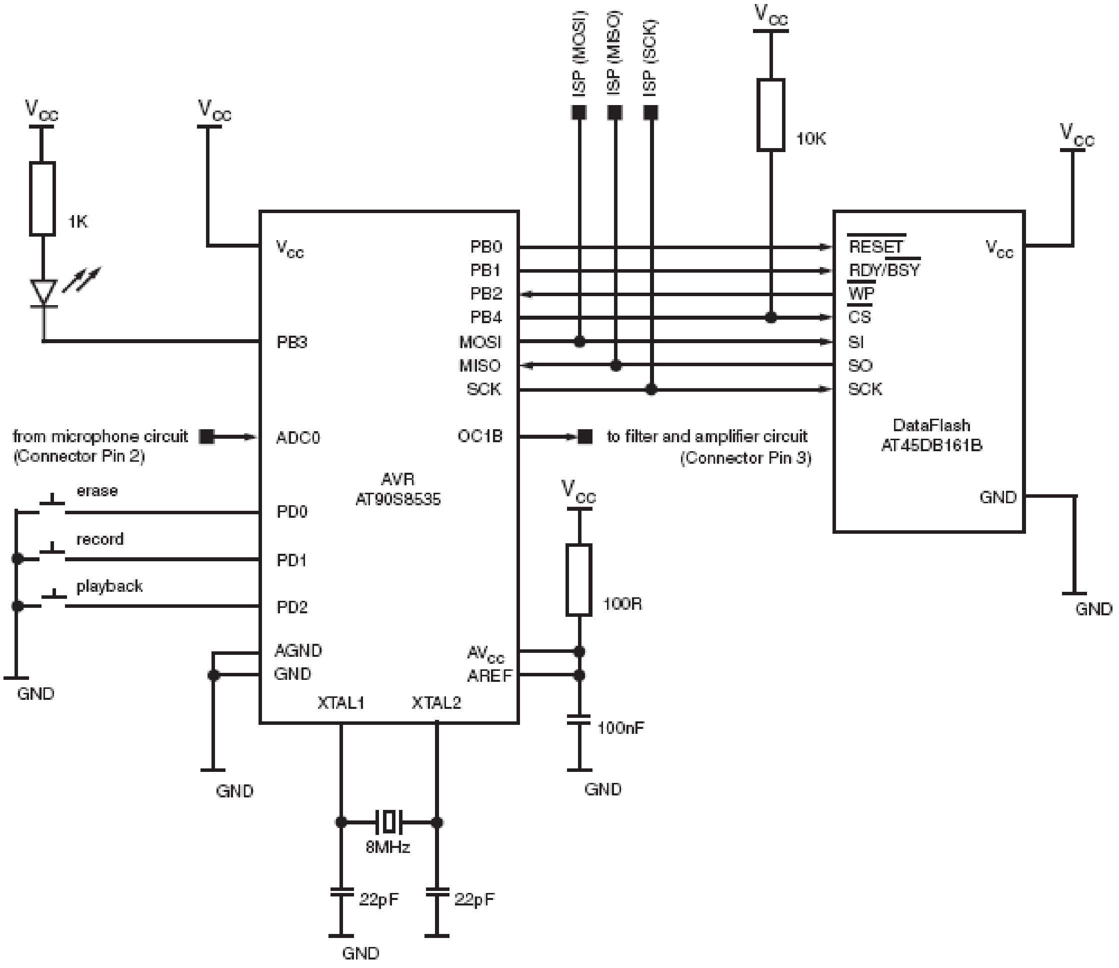

Fig. 2 – Circuit Diagram for the Voice Recorder

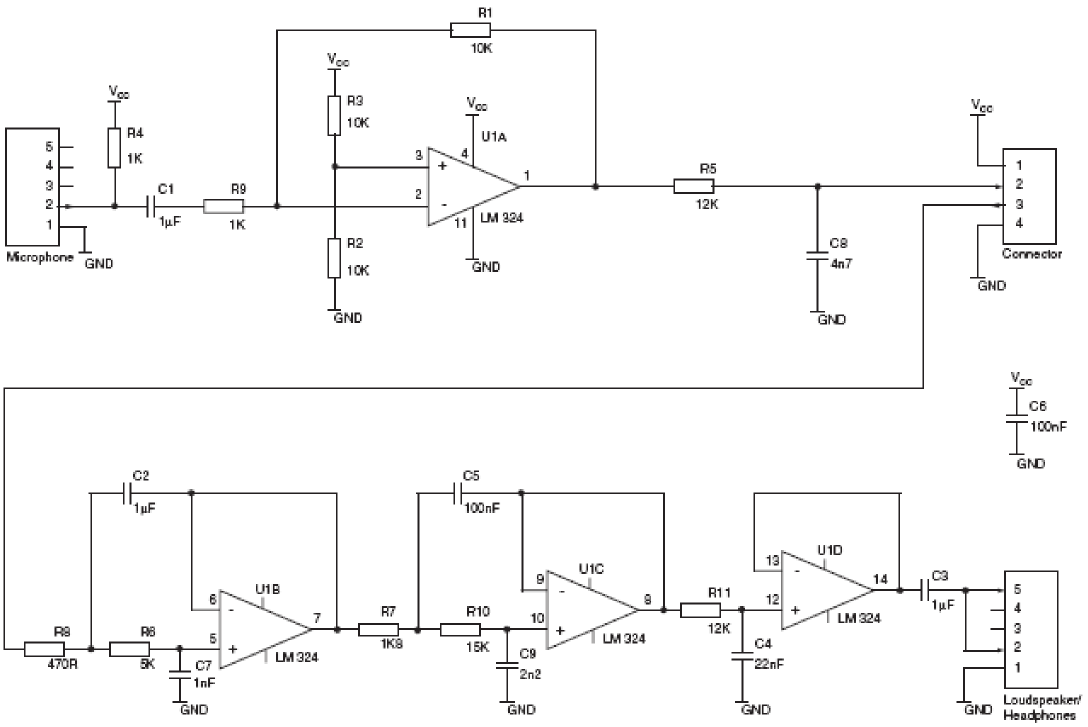

Fig. 3 – Circuit Diagram for Microphone and speaker circuit