Home Alarm System

Digitala projekt (EITF40)

Supervisor: Bertil Lindvall

Written by:

Christer Andersson (dt07ca5)

Alexander Nässlander (dt09an2)

3/5/2013

Abstract

In this report we will show our project of a home alarm system, created in the course:”Digitala projekt (EITF40)”. It´s a simple but yet a good system running on an AVR ATMega16. The system can handle up to 8 different digital inputs from sensors and up to 3 different analog inputs. Our system uses an IR-detector as an analog input that reacts on a temperature difference of 5 degrees Celsius. “Easy-to-use”-interface with a display and a keypad. It supports two different modes of alarm, day alarm and night alarm. The system can withstand power failure and recover to its former states.

Product design specification

To determine the product design specification we started to look at a real alarm system. We asked ourselves: “Which features does a real alarm system have?”. After some discussion of whether it is possible within the time limit or if the system could handle all the components that we wanted, we came up with the following list of requirements.

Requirements

- Handle 8 digital sensors, 4 timed and 4 untimed and 3 analog sensors.
- Analyze the analog signal and compare it with a reference.
- When the alarm is on a green LED will glow. When an alarm is triggered a red LED will flash. To indicate that the user has armed the system a LED will be flashing in green and red colors. A green LED will flash to warn the user that the alarm will go off within some seconds.
- Turn on and off the alarm with a numeric keypad, by a 4 digit numeric code.
- The user is able to change the 4 digit numeric code.
- Possibility to choose between night and day alarm.
- The user is able to change the arming/disarming time of the system, between 10 to 60 seconds.
- If an intruder is detected and the system crashes due to a power failure of some kind, the system will “remember” that a trespass has occurred and warn the user.
- A keypad for the user to affect the system as the user desires.
- A display must be integrated to the system so that the user is able to affect the system easier.

Components

The following components were what we used to build this project.

Processor

The processor we used where the 8 bit AVR ATMega16, that can be run on speeds up to 16 MHz, but then it needs an external clock. Since we did not feel that there would be a need for more than 8 MHz we settled for the internal clock. In the memory department it has 1KB built in RAM and a flash memory of 16 KB. There are 32 I/O-ports divided on four ports named A, B, C and D. Port A has a built in A/D converter.

LCD-Display

For the project we chose a 4 lines alphanumeric display with 20 characters per line. This is used for communication between the alarm and the user.

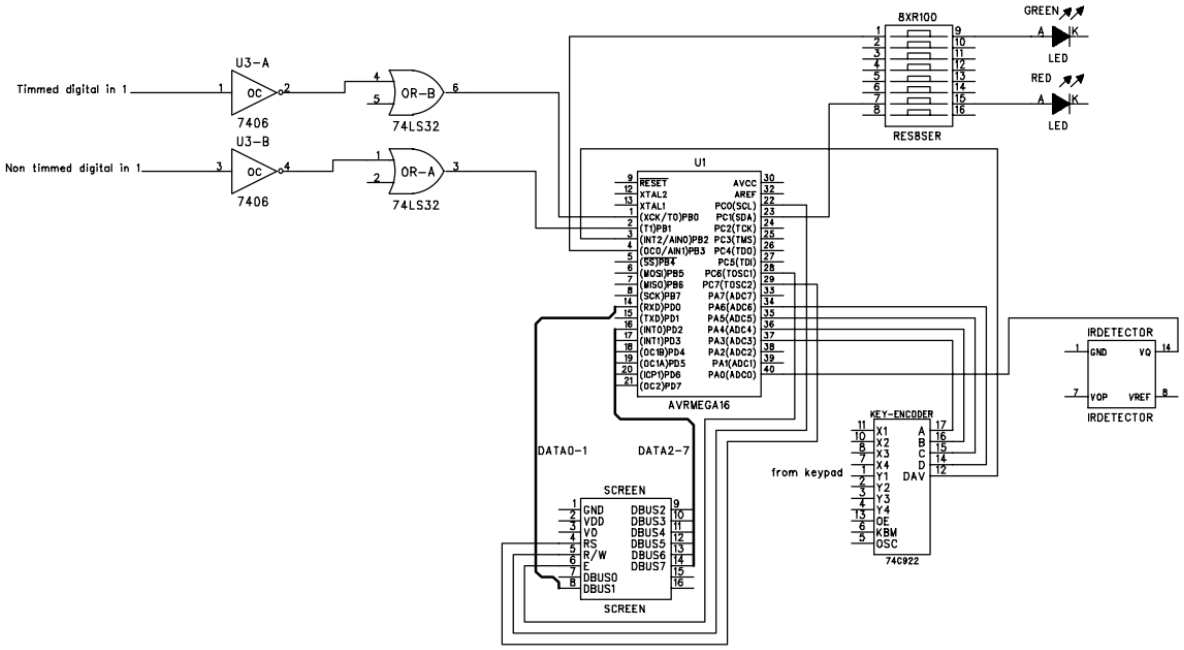
Keypad and decoder

We used an keypad with 16 keys, 0-9 and A-F, divided on four rows and four columns. To save a couple of I/O-ports on the ATMega16 we connected the eight ports from the keypad to

a decoder which had four output ports and an enable signal. The decoder reads from the keypad with a frequency that is determined by the difference in two capacitors connected to it. The decoder sends an enable signal when a button is pressed and the data is available for the processor.

Magnetic switch
Works like a normal switch, when closed it leads electricity and when opened it does not.

IR-detector
The IR-detector reacts when the temperature differ more than 5 degrees Celsius from the environment. It then changes the voltage on the output pin, higher for higher temperature and lower for lower temperature. Slow temperature changes are not measurable since it calibrates itself back to the standard voltage for stable temperatures. The standard voltage is: ——— and it ranges from 0V to Vop, which in our case is 5V.



Figur 1: Circuit diagram

The implementation

One of the original ideas was that the system should be able to communicate with a computer or server via an Ethernet jack. In order for the system to use the Ethernet jack, we had use 5 pins of the ATMega16. At the moment we could not afford a solution with an Ethernet jack because the lack of pins, so we changed the diagram of the digital sensors (the magnetic switches). There are simply two ways we handle the digital sensors in our system, either the magnetic switch triggers the alarm right away or the alarm is triggered after some seconds. That´s when we came up with the idea to either use two logic OR gates or one MUX for the 8 magnetic switches. Since two OR gates only consumes 2 pins and a MUX consumes 4 pins

we decided to use two OR gates. We used them in our design as follows, one OR gate for the magnetic switches whose signals triggers the alarm right away and one OR gate for the magnetic switches whose signals triggers the alarm after some seconds. This way we only need two pins for 8 magnetic switches.
In order to save some more pins on the ATMega16 for the Ethernet jack we used a decoder for the keypad. Instead of using 8 pins on the ATMega16 we only had to use 5 due to the decoder.
By applying these methods and components we had enough pins for the Ethernet Jack, one can read more about this under “Future ambitions”.
To get a better understanding of the IR-detector we hooked it up with an oscilloscope and watched the output voltage differ when we passed by it. By doing so we found appropriate values for our system, that we later used as constants in our code. One can find these constants in our code.

Software

Our project is developed in C using AVR studio 4 and Notepad++. After the system is initiated it goes into a while loop and checks for changes to the alarm signals. Meanwhile any interaction with the system is done via the keypad which sends an interrupt to the processor. When this happens the system goes into the interrupt code that handles any key press and does the corresponding task. Then it goes back to the while loop checking for changes. Some of the variables are saved in the EEPROM to coop with the requirement that the system should stay in the same state after a power out. For example if the alarm was triggered and the power goes out the system should still be triggered when power comes back. The code is also saved in the EEPROM to guarantee that the code stays the same.

Result

The system works to specifications that we wanted, even thou we discussed some things during the project that had to be put as future ambitions due to time constraints. We had some problems during the implementation and learned quite a lot, some in software but mostly in hardware. Time was one problem that we did not solve and decided that the solution for us where to use the time handling from the Ethernet connection in a future implementation.

Future ambitions

There are a couple of things that we would have liked to do that the time did not allow for. First and foremost is the connection to a computer or server via an Ethernet cable. And if that were finished that would have opened up for the possibilities to implement a list with time and date for when the alarms had been triggered. Lastly using a MUX for the magnetic switches would have given us the possibility to say which sensor that triggered the alarm.