ATxmega256A3BU-XPLAINED
Simon Says
Push-LED Game

University of Louisville
ECE 412 Introduction to Embedded Systems - Final Project
Dr. Harnett - Fall 2014

Team Celadon

Taylor Hans
Ricardo Benitez
Robert Cook
Austin Schroder

Abstract

The goal of this project was to learn about using transistors, interrupts, and LEDs combined. Our final project consists of creating a simple push-LED Simon Says game. This game is essentially a memory sequence game, where the user watches a developing LED sequence, and then has to input it correctly to move onto a more difficult sequence with more elements to memorize / faster. As a whole, this project combines much of the knowledge of embedded systems and skills we have learned throughout the semester. Our design consists of a few key components that, while distinct and separate, must work together to allow the overall design to function. On the software side, the code represents the "brain" of the memory game. It is the code that decides the sequence of lamps to light and acts upon the user input, deciding what to output based on input. The events are interrupt-driven, and certain user inputs trigger an interrupt handler that decides what happens next. The hardware side of the project consists of interfacing the a3bu board with the push-button lights, which are, in turn, used to interface with the user. The output portion of the circuit must raise the voltage level for the lamps, and the input circuit must provide voltage to the board input when a button is pressed. Ultimately, a working Simon game must have both the hardware and software components working perfectly.

The equipment used is: 4 light-up pushbuttons (12-volt LED), Atmel AVR atxmega256a3bu microcontroller board, 4- PN2222 transistors, soldering iron, lab voltage source, assorted wires, JTAGICE3 debugger, Personal Computer, Atmel Studio 6.2. The platform used is Microsoft Windows 7. We experienced a problem with the JTAG connector pins which caused one a3bu board to "brick."

The goal of the project prompt was achieved. We successfully combined hardware and software in our final working version.

EEPROM

ATxmegaA3BU-XPLAINED evaluation board's EEPROM was utilized to store and keep track of the game's high-score in a non-volatile way so that when the device is powered down this data will be permanently stored. There was also an implementation of clearing this value via an interrupt by pressing the RED LED when the game is in its default state menu mode.

Interrupts

There were four basic interrupts configured and set for the game. Each one being the corresponding push-LED user input switch in the game's circuit.

Software (Program)

Simple binary / boolean value, integer variables were used in the main C program global stack flow to control the execution of the game. They were used to control waiting for user input into the interrupt service routines (ISRs) for the LEDs as well as in design for a menu system. Another integer variable was used to interpret which last input LED was pressed and check that the sequence was read-in, entered, correctly in by cross referencing it with an array containing the entire elements.

One thing that is very important to note, and caused a lot of problems in designing this game, is that the ATXMega board does not like nested loops, and will typically ignore the nested loop entirely, leaving out important code that your game relies on. Especially with game code, using timers and comparisons constantly, this was information that was invaluable once it was discovered.

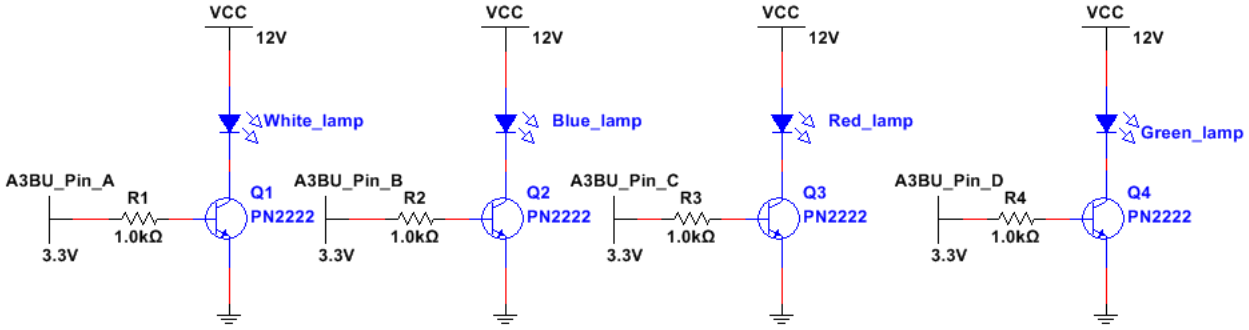
Source Code

The final project Atmel solution repository link is here, it contains the main program's source code as well as all included ASF dependency libraries bundled up inside.

GitHub Project Repository

Circuit Schematics

Output:



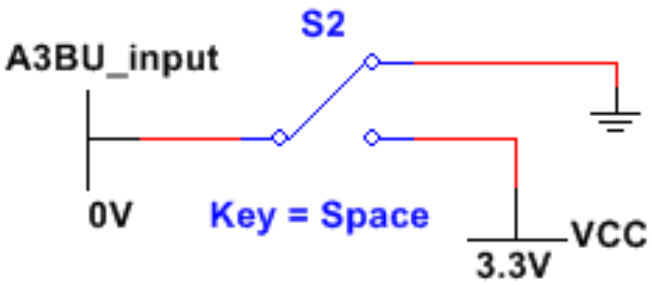
PN2222 transistor consist of three pins: base, the one in the middle, the emitter, in this case the one with the arrow going to ground, and the collector, which is where the lamp is connected. The base has a 1 kΩ resistor which is there to protect the transistor from excess current, the lamp receives 12V and then is connected to the collector of the transistor. Each lamp consists of an LED and a resistor (the resistor is inside the LED). The lamp operates on 12V. When a voltage is applied across the leads of the Light Emitting Diode, the depletion zone of the diode shortens, and current flow is allowed. The LED emits light depending on how much voltage is applied, and therefore how much current is flowing. If a voltage of, say, 3.3 V is applied across the LED, a small current will flow, and the light will be very dim. In order to have the brightest, clearest illumination, we provided a full 12 V to the LED. We realize that in this configuration

(RTL) the transistor gives the A3BU control over the lamp; if the 3.3V is set to ground (or disconnected) the lamp will not turn on, so it is basically a switch that is activated by the 3.3 V.

In the schematic above, it's listed as pins A through D, when in reality, we used PORTC, pins 1 and 2, and PORTD, pins 0 and 1. We had to use two different ports, because it was the only way we could determine which button actually caused the interrupt. Otherwise, each button looked identical. So, we used both interrupt setups on ports C and D, INT0 and INT1, and had an interrupt service routine for each interrupt setup, giving us which button would trigger an interrupt.

Also a good side note that applies to any project, do NOT use PORTB pins 3 through 7 for interrupts. This is the JTAG interface for the board, and instead of writing new code, the board will run the ISR's located within the new code, and not allow the board to be wiped. This essentially bricks the board, save for what we're assuming would be a factory reset.

Input:



The switch that we used has a total of three terminals, one of them is connected to ground, the pin below the previous one goes to a 3.3 VDC and the last one connects to the input pin of the A3BU. The two pins (one connected to ground and the other one connected to 3.3V) sets the normally open (NO) and normally closed (NC), in our case we have a normally open which means that the switch will send a signal to the A3BU when the switch gets press down.

Conclusion

In conclusion, our Simon game was a success. The goal was to test our knowledge of embedded systems in designing a working project. We were commissioned to conceive of, design, build, and test a product. Thus, much about the creative design process of engineering was learned. In addition to the raw electronics information learned, we gained experience working as a team. In the real world, large-scale projects require many different engineers to collaborate. So, ideas must be shared clearly and effectively in order for every engineer to be on the same page. A project can only move forward if every contributing piece is functioning properly. As Team Celadon, we communicated our ideas effectively, and this was instrumental to a successful project.

References

- PN2222 Datasheet
- Dr. Harnett (University of Louisville)
- Electronics HW 10, RTL Part
- Sparkfun-Switch-Basics
- Sparkfun-Transistors
- AVR-Interrupts
- 8/16-Bit Atmel XMEGA A3BU Microcontroller - ATxmega256A3BU
- XMEGA A3BU XPLAINED Design Documentation
- 8-Bit Atmel XMega Microcontroller AU Manual