Celadon Simon Says Push-LED

University of Louisville

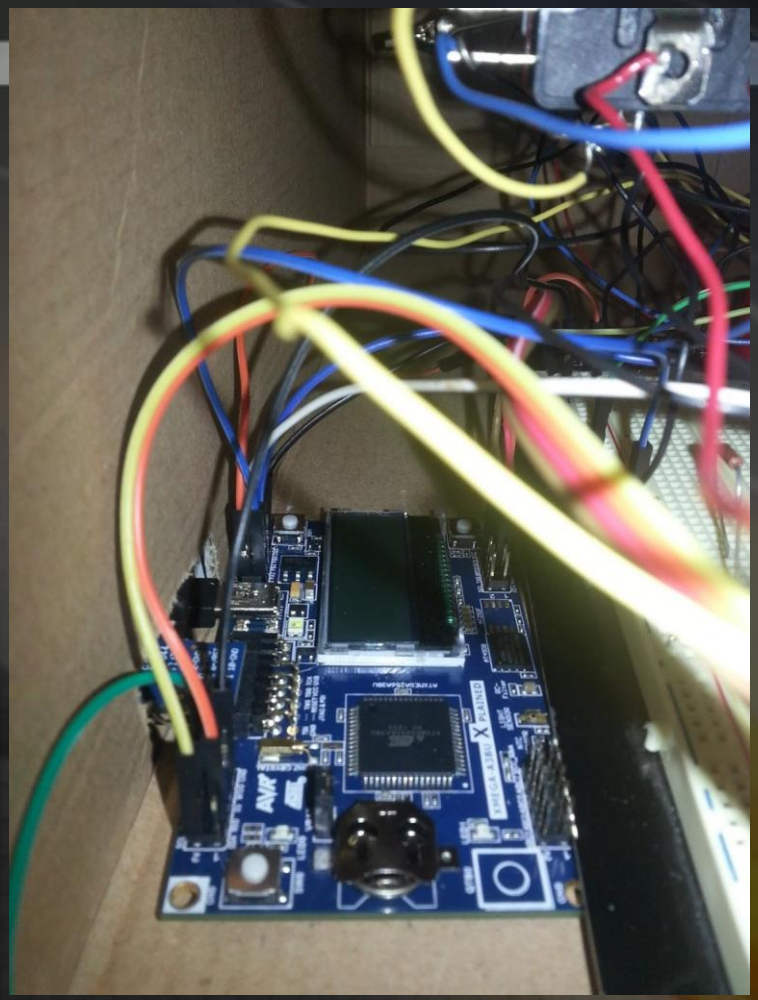
ECE-412 Fall 2014

Taylor Hans, Ricardo Benitez, Robert Cook, Austin Schroder

Game: Simon Says

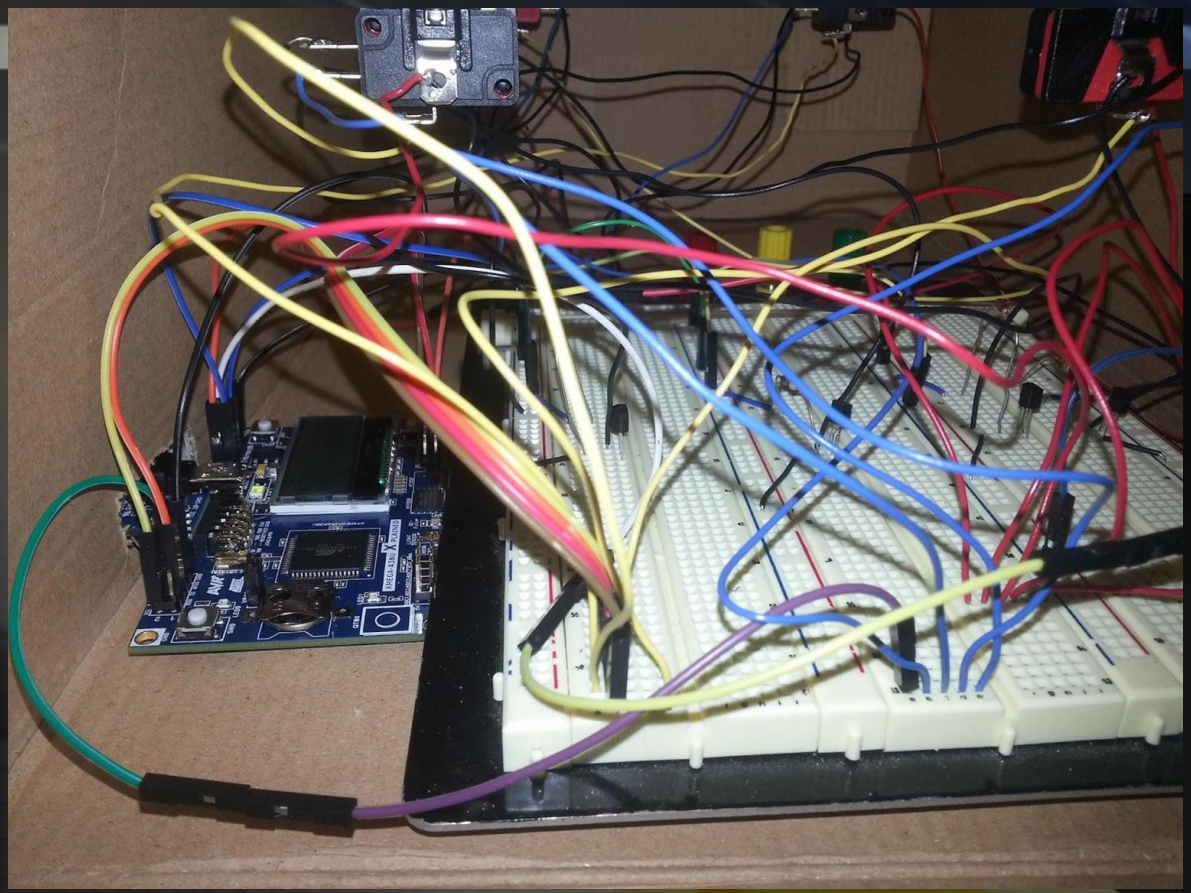
- User follows, memorizes, and attempts to enter correct push-LED sequence
- LEDs are connected to J2 PORTA pins for generic output of the sequence, and the inputs are connected to J1/J3 PORTC and PORTD pins
- No sound (YET!)

Under the Box: Interface



This shows the interface setup for the XPLAINED J expansion headers and PORTs used for both input and output connections for the game

Under the Box: Circuit

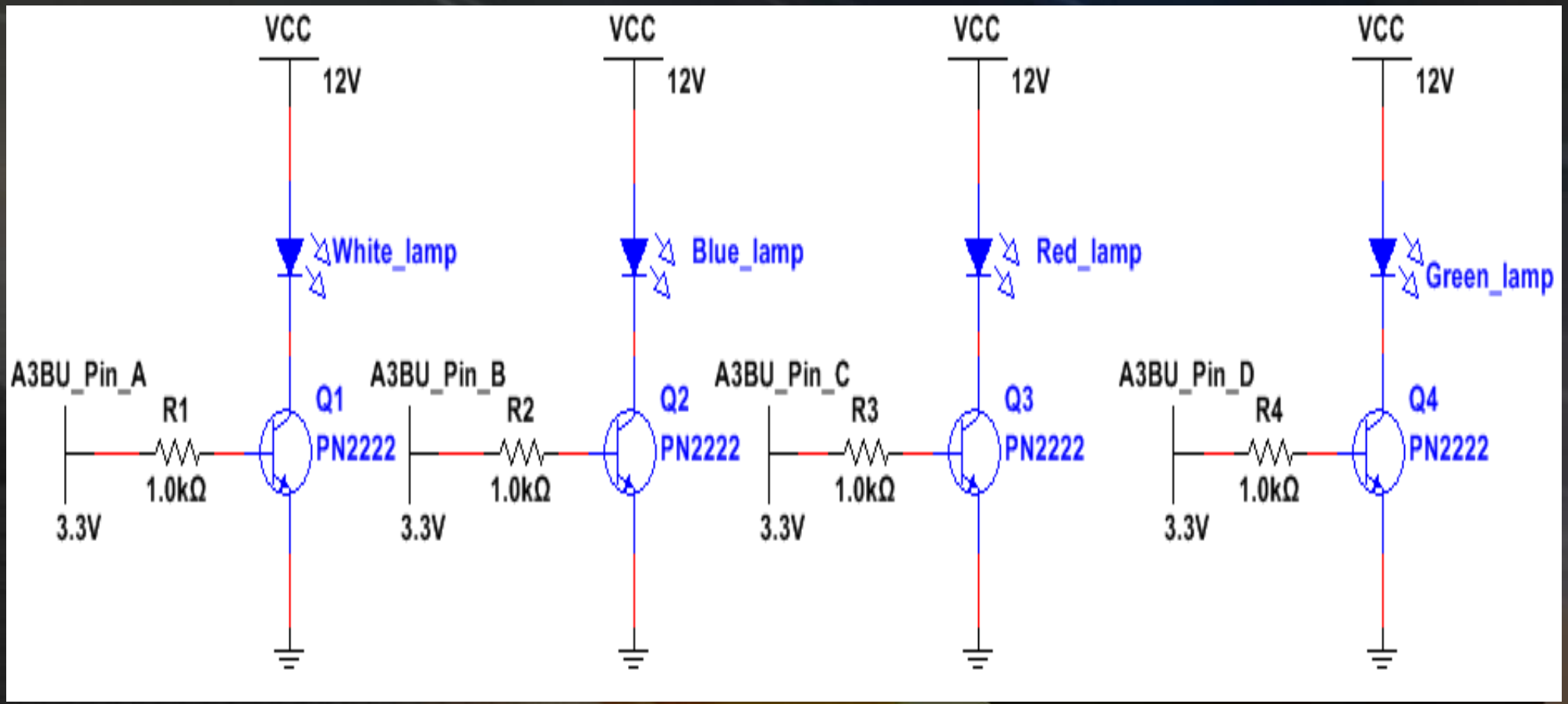


Most of the loose wiring was due to needing to pull the A3BU pins to somewhere else, something that’s not easy to do tightly.

Under the Box: LED / Switch

- This shows the wiring between the LEDs, switches, circuit, and bread board
- It is directly underneath the top of the box where these are hooked up
- Soldering, and a last minute clamp, was used to hold connections securely in place

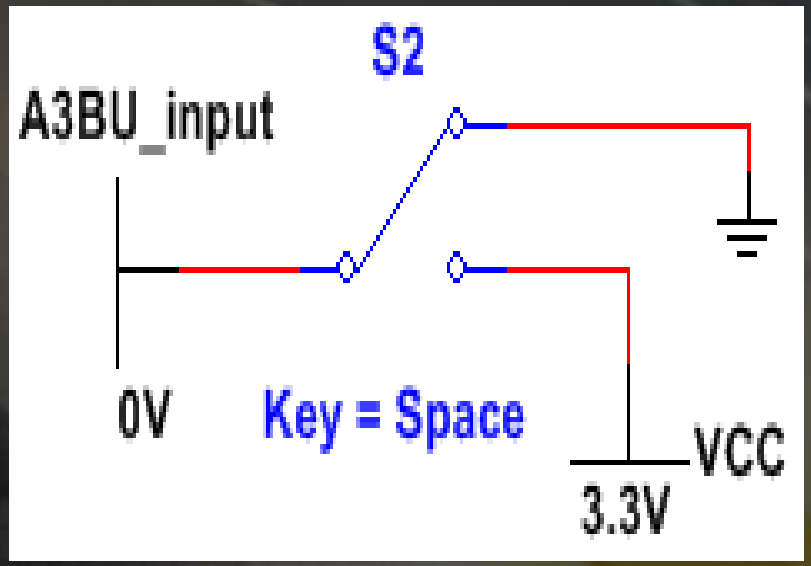
Schematic: Output



Schematic: Output/Input

- For the output:
  - In this configuration, the Transistor behaves as a switch, completing the circuit when 3.3V is present at its base and turning the LED on. PN2222 transistors saturate when Collector/Emitter voltage is 1V.
  - The LED and its 12 VDC power supply are connected to the Collector of the transistor and the emitter goes to ground
- For the input
  - The switches consist of three terminals: NO (normally open), NC (normally close), and common.
  - The common is connected to an input of the A3BU, normally closed goes to ground and normally open goes to 3.3V
  - Pressing the switch completes the circuit and it will send a signal to the A3BU

Microcontroller / Switch



- Switch is momentary contact, Single Pole Double Throw (SPDT)
- Normally Closed (NC) contact between A3BU input and ground
- Normally Opened (NO) contact between A3BU input and VCC

Finished Product: Debugging

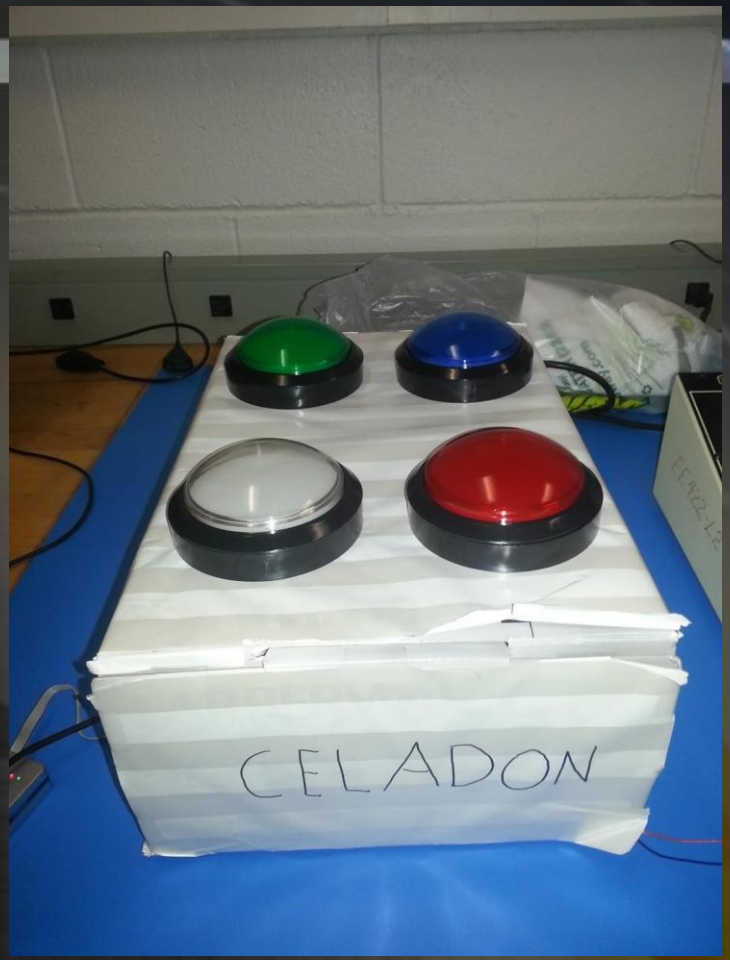
The box allowed us to hide the wiring, plus an organized method for using the JTAGICE3 debugger to completely finish the game's software

Problems Shortlist

- Ran into a problem where the board wouldn’t accept nested loops/ifs - Would just ignore the loops or statements within the outer loop. Never did find a solution, but we moved the inside code to an external function, and it worked fine.
- High score always tracked as double the correct number of points - We just halved the number of times the lights indicate a point to solve that problem.
- At first, our interrupts wouldn’t let us determine which button caused the interrupt, so we spread the inputs across two ports so each button would have a different interrupt service routine, giving us exactly which button caused the interrupt.
- A3BU got bricked when we tried to use PORTB for interrupts, which happened to be the same PORT the JTAG interface uses.

ZOMBIE LIGHT!

Finished Product: Instructions



- In order to start the game push the green button, all the lights will turn on indicating that the game is about to start
- The white button will flash the highscore (non-volatile through EEPROM)
- The red button will reset the highscore
- The blue button is to make the lights go crazy

Video

References

- PN2222 Datasheet
- Dr. Harnett (University of Louisville) and her Electronics HW
- GitHub Project Repository
- 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

Questions?