Electrical Recruit Project: Brightness-Controllable Arduino LED Board

For SDSU Mechatronics

System Purpose

The system is a PCB that contains an ATMega328P-A microcontroller, a pushbutton, and an LED. The brightness of the LED can be switched on and off via the button, and have its brightness set to various levels through the Arduino serial monitor. It supports ignoring serial input when in the off state, 6 different brightness levels, and brightness readout when setting the brightness through entering letters into the serial monitor.

System Explanation

The brightness of the LED is switched on and off through a pushbutton that works by allowing a 5V signal to enter the Arduino digital pin 7. Pushing the button triggers a LOW or HIGH state, and by using the if statements in the first half of the Arduino code it will set the PWM pulse rate to 127 (half brightness) if it was at 0 (off state). If the value was ever higher than 0, it will reduce the brightness to 0 instead, turning it off.

The serial input is ignored when the switch is in the off state, done by applying a check for the last button state for each of the serial input if statements that control brightness levels. The last button state is changed in the first half of the code where it checks the current brightness, and if it is higher than 0 (indicating that it is on) it sets the state to off, and vice versa.

The 6 different brightness levels are done by using analongWrite() to change the pulse width of the incoming 5V signal, with higher values increasing brightness and lower values decreasing brightness. The bigger the value, the bigger the width of the 5V signal, and the higher the brightness. There are 6 if statements tied to a distinct character that, when entered into the serial monitor, sets the brightness to a certain level by using analogWrite().

Challenges Faced

Much of the challenge of this project was learning how to use KiCad, such as choosing components and placing them. This just took time for me to get accustomed to using the software. Another challenge was trying to route all the traces in the PCB editor without any colliding, which was averted by using two different PCB layers with two layers of traces, and for the surface mount chip I used vias to switch between layers to go under traces and to get to their pins without collision.

Conclusion

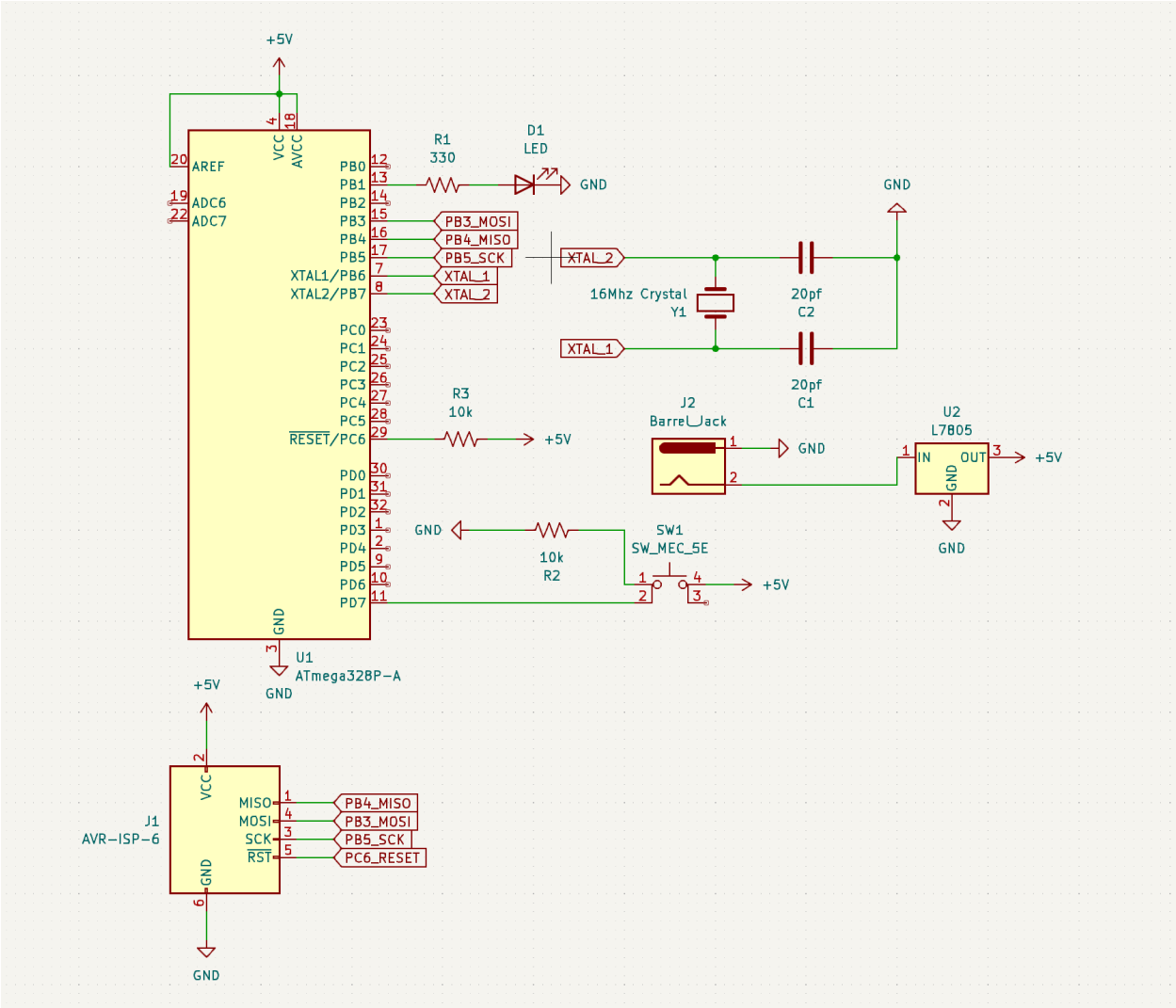
Overall, this was a very fun project that introduced me to the proper PCB design workflow, and also taught me many things about how circuits work such as the relationship between current, resistance, and voltage.

BOM

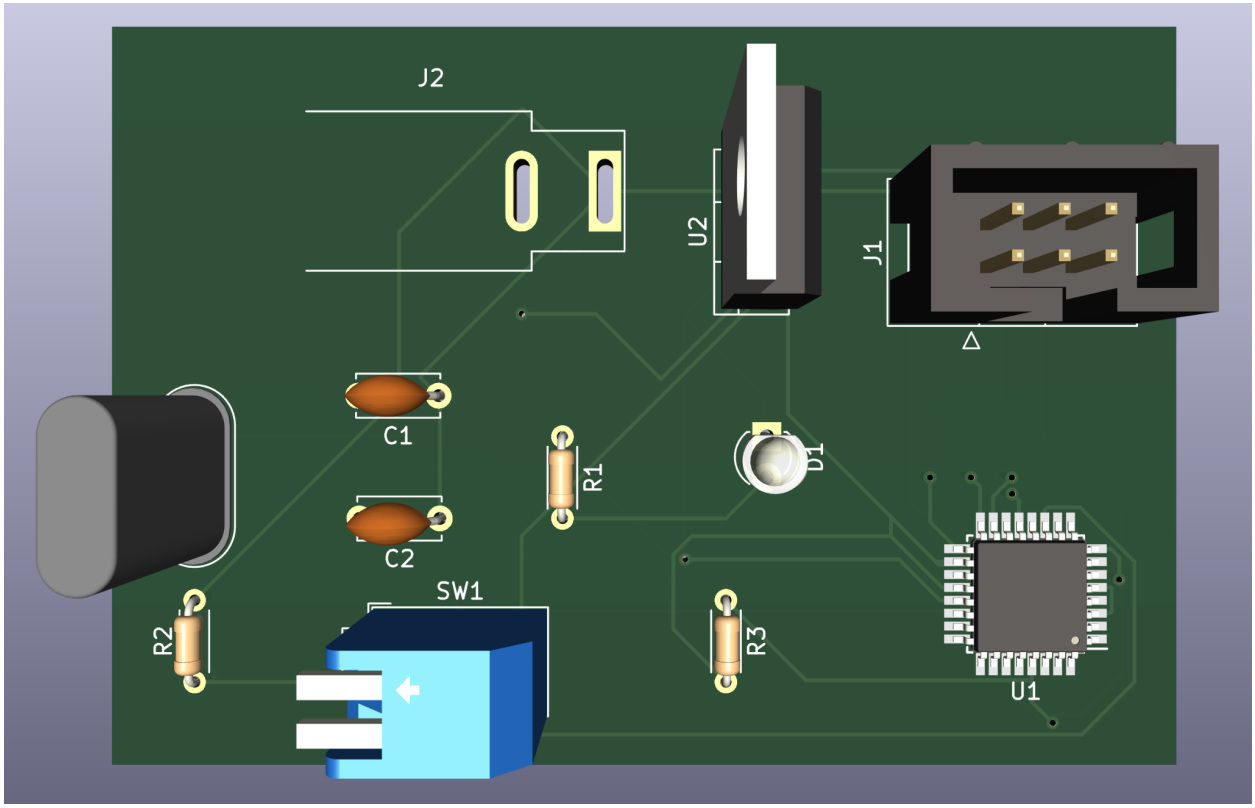
Parts	Quantity
20 pF Disc Ceramic Capacitor	2
Clear LED	1
AVR-ISP-6 Header	1
Barrel Jack	1
Resistor (330 ohms)	1
Resistor (10k ohms)	2
SPST Piano Button Switch	1
ATmega328P-A Package Microcontroller	1
LM7805 5V Regulator	1
HC49-US 16Mhz Crystal	1

Total price for an equivalent of 8 GoBilda kits was $1,599.92.

Schematic


+5V
VCC
AVCC
AREF
ADC6
ADC7
PB0
PB1
PB2
PB3
PB4
PB5
XTAL1/PB6
XTAL2/PB7
PC0
PC1
PC2
PC3
PC4
PC5
RESET/PC6
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
GND
U1
ATmega328P-A
R1
330
D1
LED
PB3_MOSI
PB4_MISO
PB5_SCK
XTAL_1
XTAL_2
16Mhz Crystal
Y1
20pf
C2
20pf
C1
R3
10k
J2
Barrel_Jack
U2
L7805
IN
OUT
SW1
SW_MEC_5E
10k
R2
J1
AVR-ISP-6
MISO
MOSI
SCK
RST
PC6_RESET

PCB Front



PCB Back