Lab 2 - PSoC Intro: LEDs, Square Wave, Button, Pins



Alexis Adie and Timonty Lum

ELC 343-L2: Microcomputer Systems

Submitted:

September 21st, 2017

Lab 2 - PSoC Intro: LEDs, Square Wave, Button, Pins

Alexis Adie and Timonty Lum

Department of Electrical and Computer Engineering

The College of New Jersey

2000 Pennington Road, Ewing, NJ 08618, USA

(adiea1, lumt1)@tcnj.edu

# I. Introduction

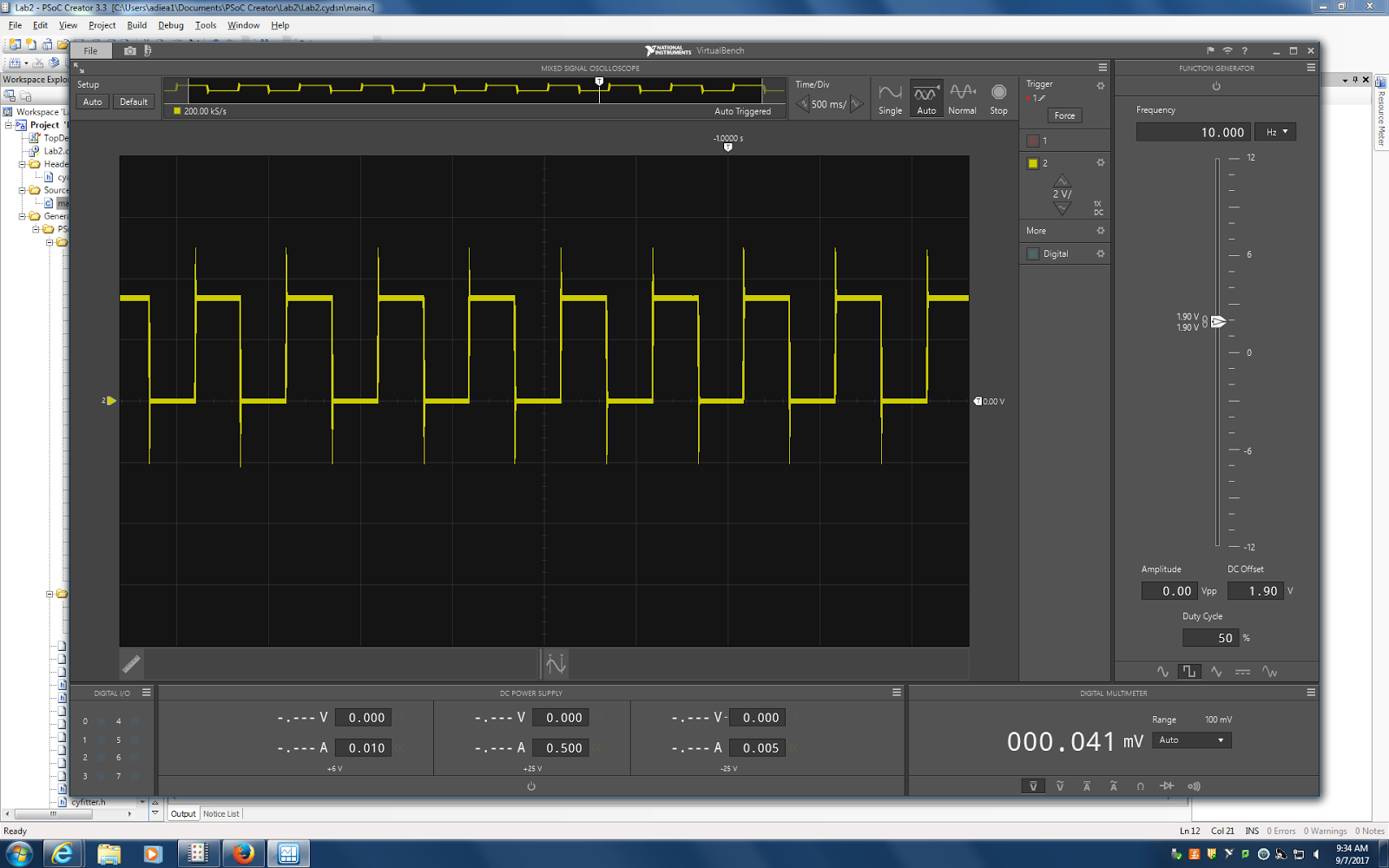
The purpose of this experiment was to us PSoC Creator to create a square wave at a designated output pin and blink an LED by sending the signal to its corresponding pin. The team was also required to read the state of a pushbutton and drive its state to a pin (LED).

# II. Methodology

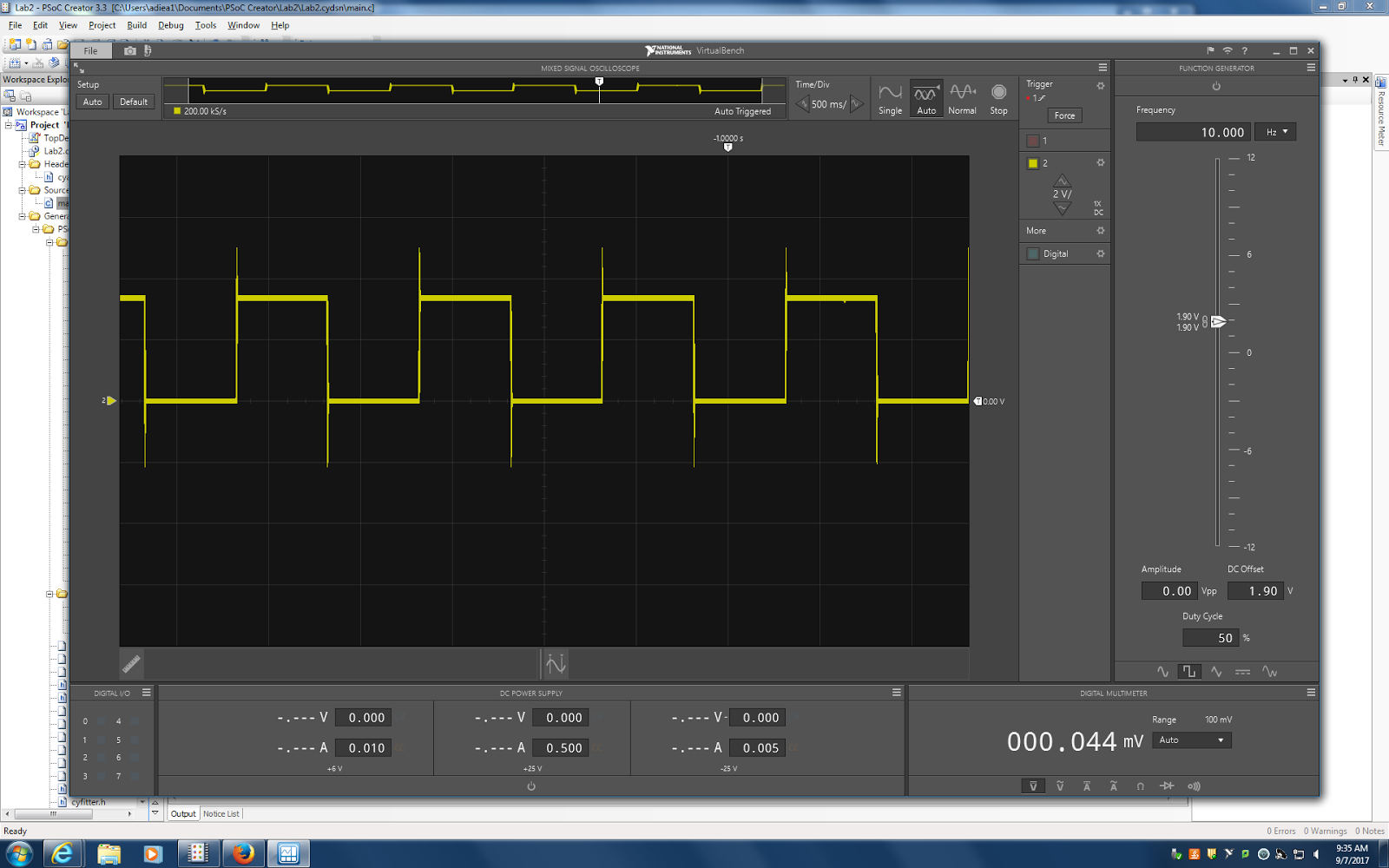
*Part 1: Producing a Square Wave on an Output Pin*

The team began by creating a new project file in PSoC Creator. As instructed, a digital output pin was added to the schematic. For the code, it was determined that setting the pin to one, delaying it, then setting it to zero, and delaying once more would create the desired square wave.

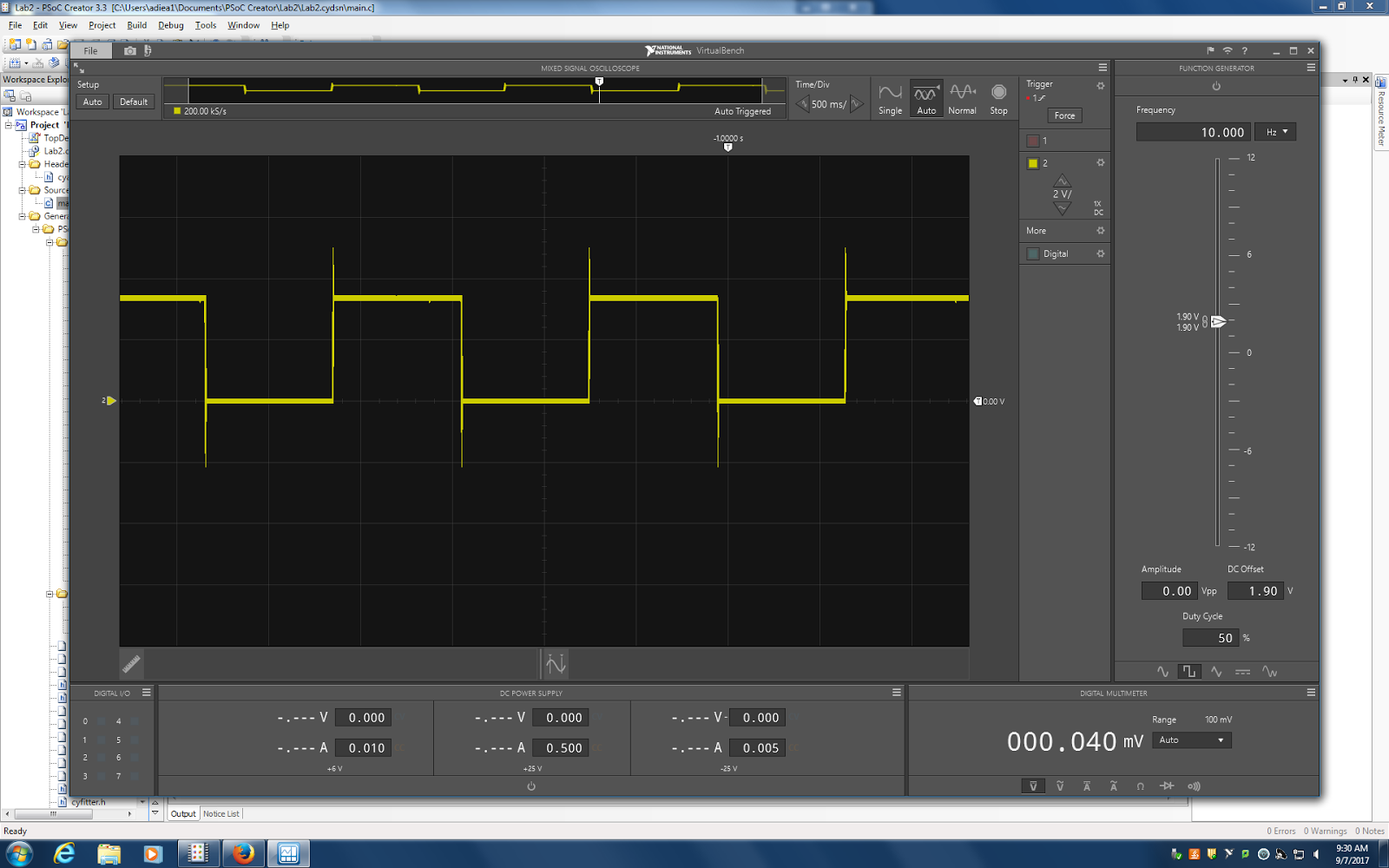
To test the functioning of the code, the board was programmed, and the circuit was tested with an oscilloscope. The team varied the time delay to 250 ms, 500 ms, and 700 ms in order to display the functioning of the square wave.



**Figure 2.1 : Square Wave Produced by 250 ms Delay**



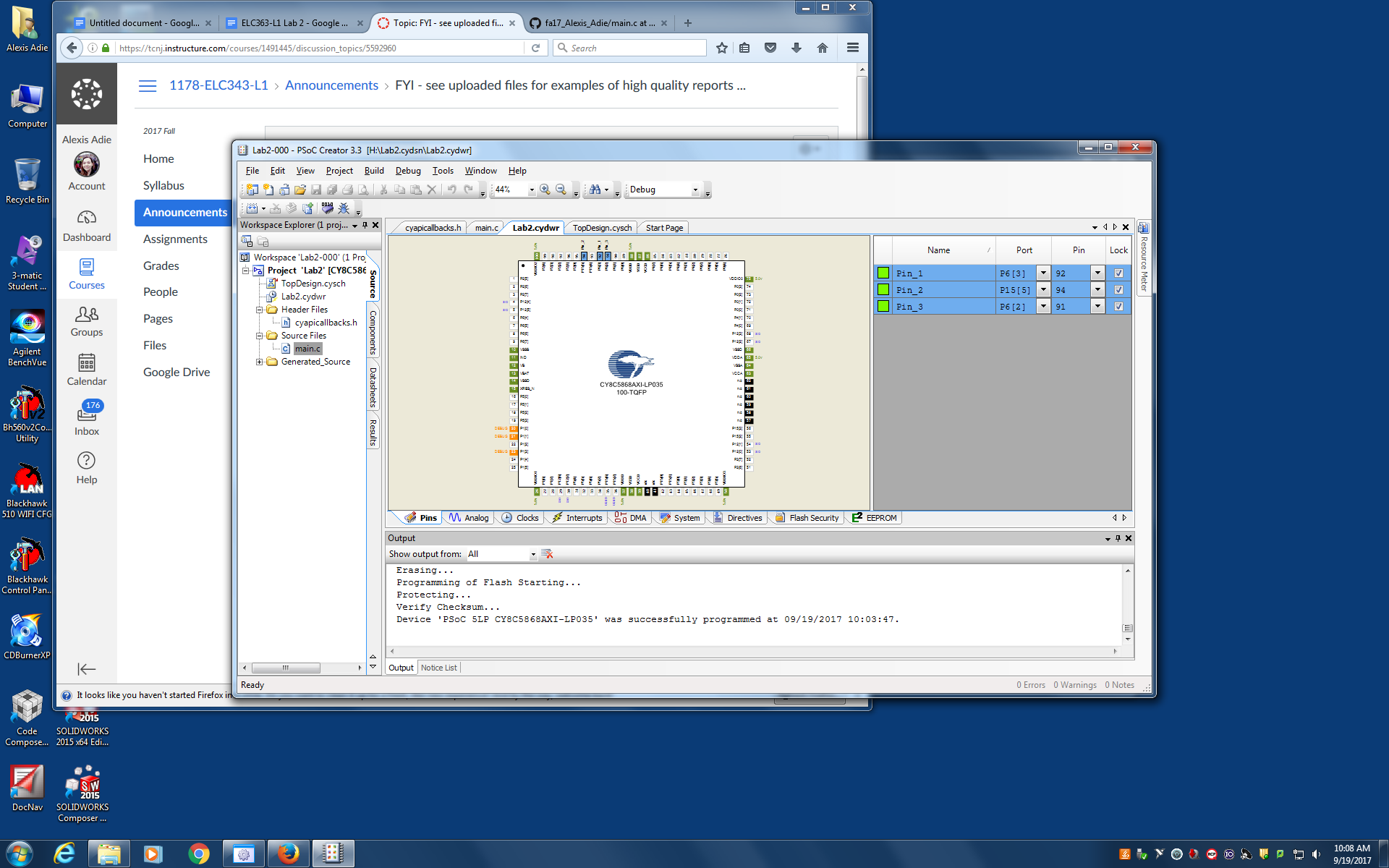
**Figure 2.2 : Square Wave Produced by 500 ms Delay**



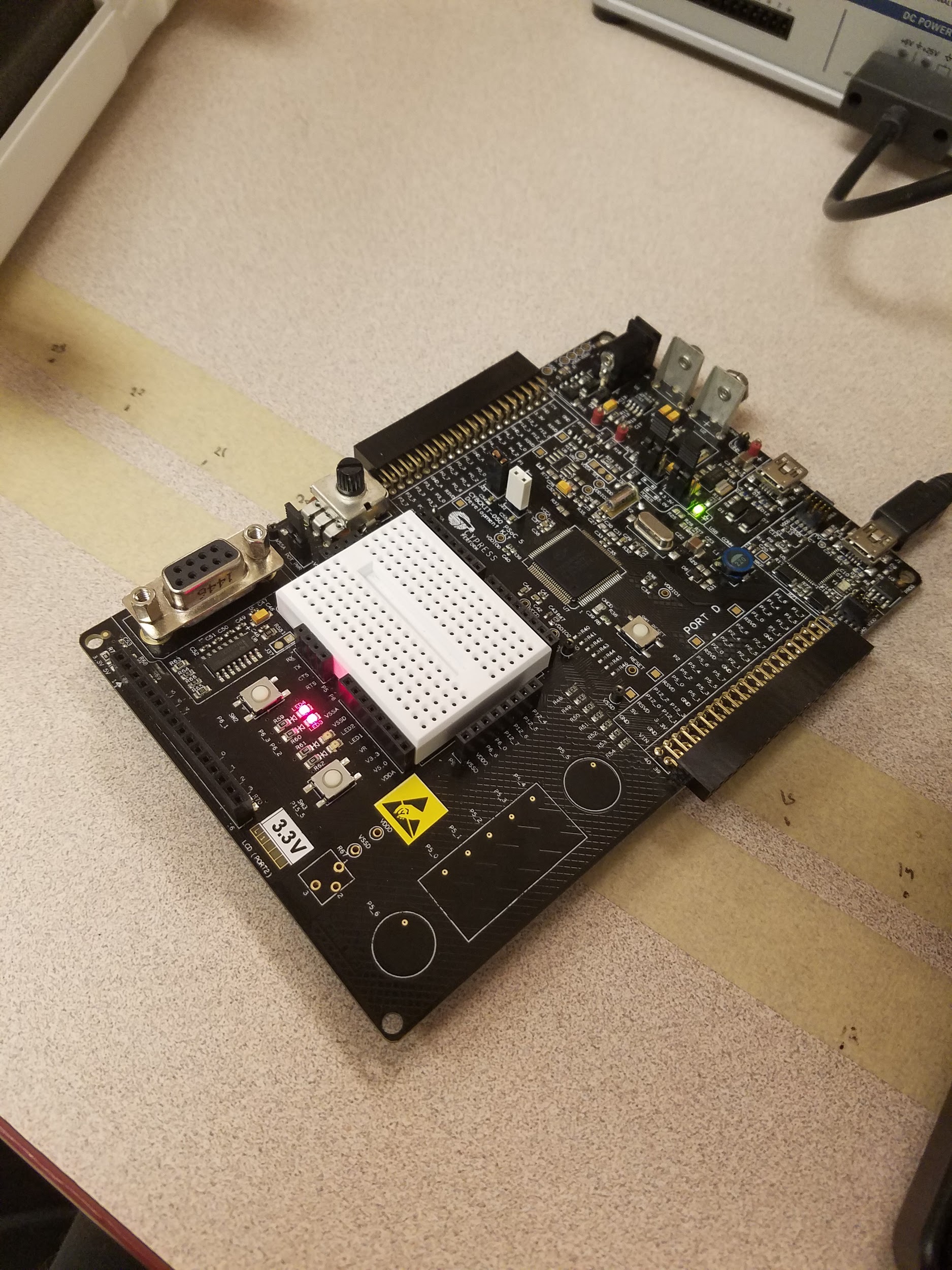
**Figure 2.3 : Square Wave Produced by 700 ms Delay**

*Part 2: LED Toggle and Blinking LED*

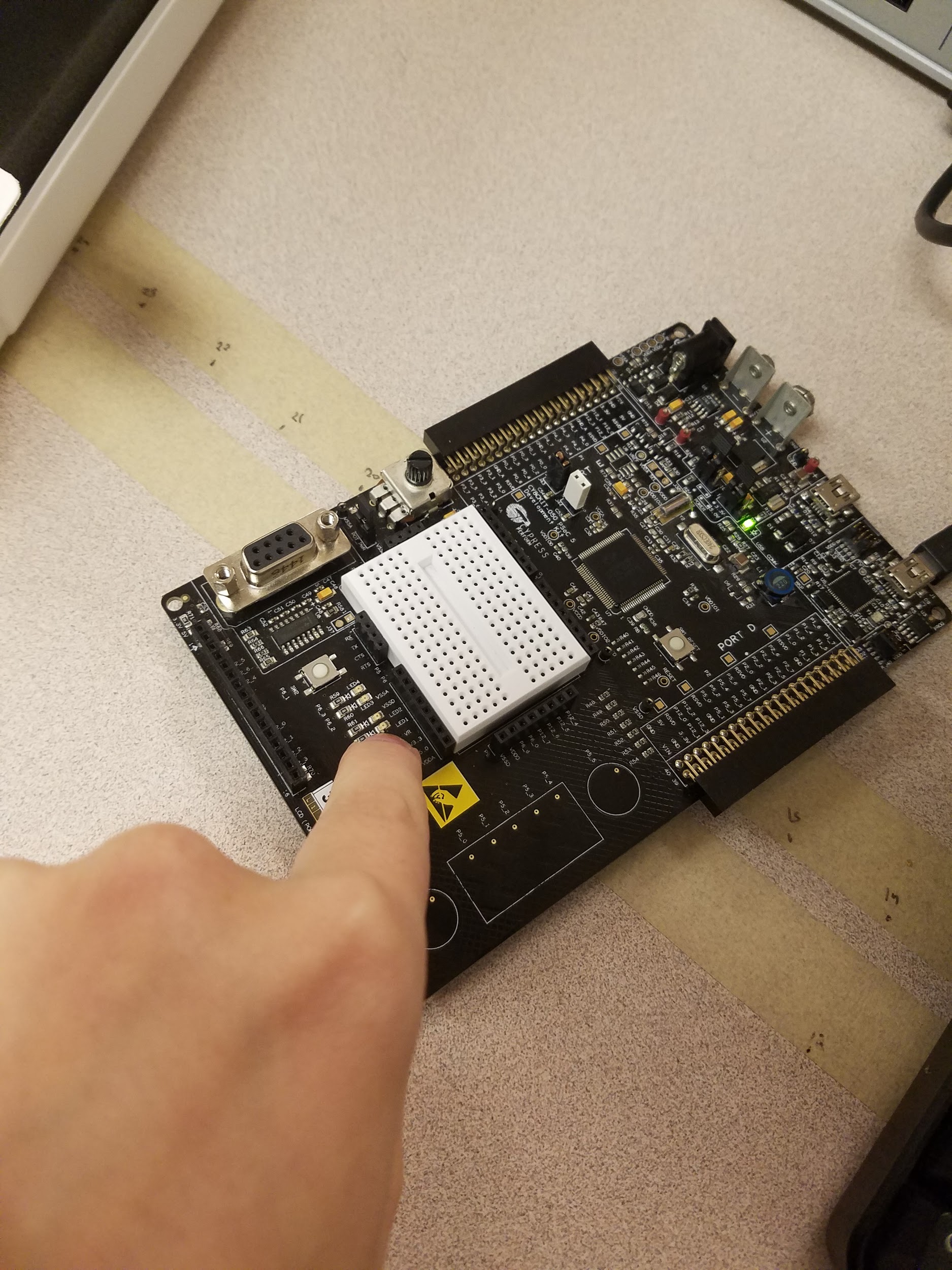
An LED was wired to an output pin to begin our extension of the first design. The code was adjusted by setting on/off values of time to 1000ms so that the LED flashes at a rate of 60 flashes per minute. A digital input pin and a second output pin is added to the schematic. The digital input pin is then mapped to the pin wired to the pushbutton. The second output pin is wired to the free LED and coded so that the LED reacts accordingly to the state of the pushbutton. The design was downloaded onto the PSoC board and tested.



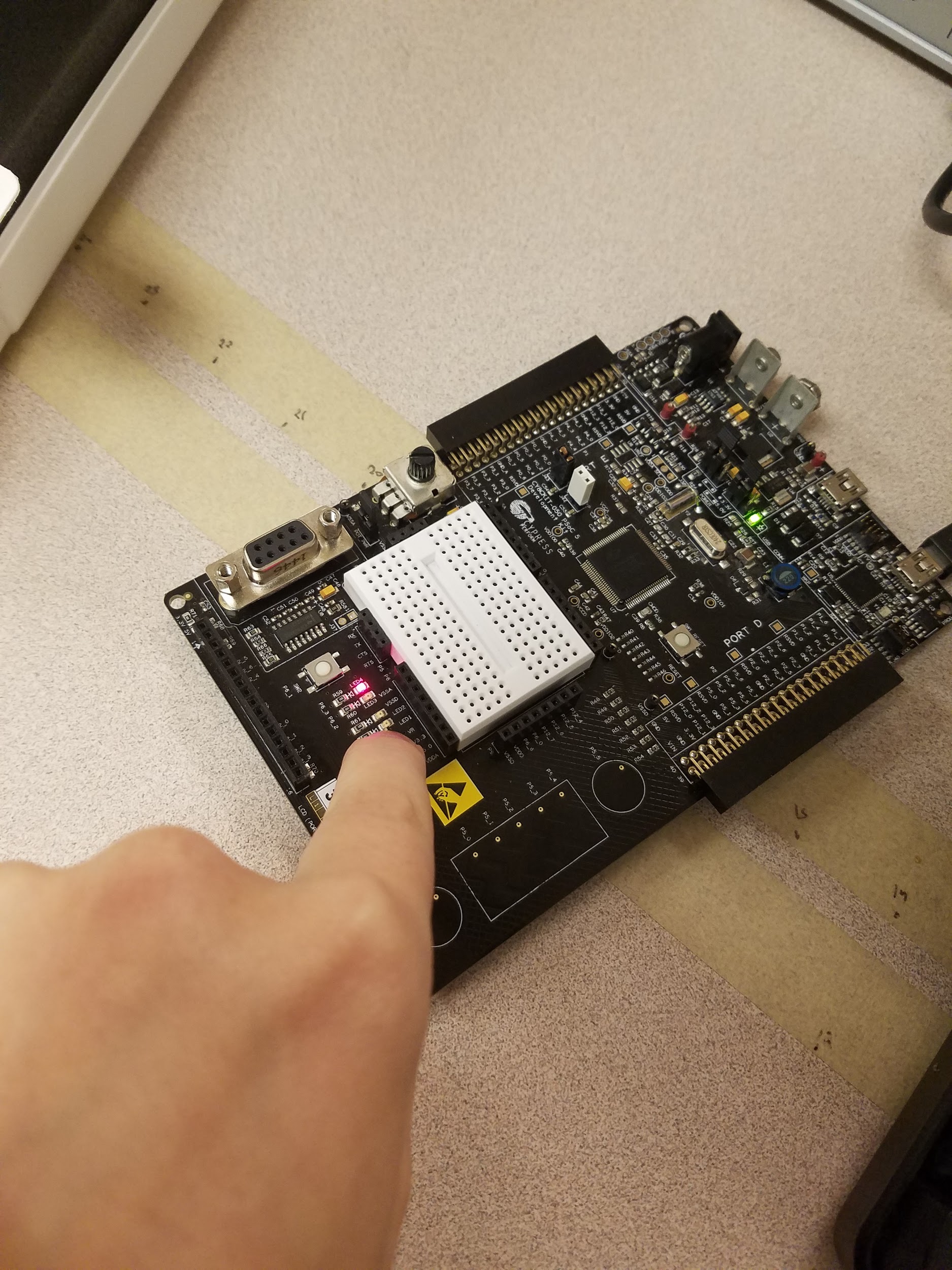
**Figure 2.4: Pin I/O Configuration**



**Figure 2.5: Blinking LED (On) vs. Untoggled Pushbutton LED**



**Figure 2.6: Blinking LED (Off) vs. Toggled Pushbutton LED**



**Figure 2.7: Blinking LED (On) vs. Toggled Pushbutton LED**

# III. Discussion

The lab required the team to use a pull-up resistor in the button toggle configuration in order to turn the static LED on and off. The pull-up resistor works by driving the current from the power source to the LED. When the resistance was “pulled up” (the button was pressed), the circuit was incomplete and as result, no current flowed to the LED.

# IV. Conclusion

The objectives to generate squares waves, blink LEDs, and drive a pushbutton state to a pin have been met with success. In Part 1, a digital output pin was added to the schematic. The output pin was referred to Pin\_1 in the C code and was configured to turn on (1) and off (0) for 500ms each. Downloading the program onto the PSoC board made it possible to analyze the square waves. The length of time of the on and off state was changed to 250ms and 700ms so check the functionality of the generator. Part 2 required a configuration on the design that made use of two LEDs on the PSoC board. The first LED used the square wave generator code from Part 1 to generate a blinking LED. The second LED was wired to a pushbutton using digital I/O pins. The C code was extended to read the state of the pushbutton and write the state of the LED according to the state of the pushbutton. After downloading the project onto the PSoC board, the first LED blinks and the second LED turns off only when the pushbutton is activated. The results of the experiment showed no deviation from the expected results.

# V. Code

/////////////////////////////////////////////////////////////////////////

Part 1: Square Wave

////////////////////////////////////////////////////////////////////////

#include <project.h>

int main()

{

CyGlobalIntEnable;

//infinite loop that produces a square wave

for(;;)

{

//Sets Pin\_1 to one and then sets it off for 500 ms

Pin\_1\_Write(1);

CyDelay(500);

//Sets Pin\_1 to zero and then sets it off for 500 ms

Pin\_1\_Write(0);

CyDelay(500);

}

}/////////////////////////////////////////////////////////////////////////

Part 2: LED Toggle and Square Wave

////////////////////////////////////////////////////////////////////////

#include <project.h>

int main()

{

CyGlobalIntEnable;

//infinite loop that produces a square wave

for(;;)

{

//Pin\_2=button so when the button is pressed

//Pin\_3 (LED2) goes off since the circuit is pulled

//and the circuit is disconnected

Pin\_3\_Write(Pin\_2\_Read());

//Sets Pin\_1(LED1) to one and then sets it off for 1000 ms

Pin\_1\_Write(1);

CyDelay(1000);

//Sets Pin\_1(LED1) to zero and then sets it off for 1000 ms

Pin\_1\_Write(0);

CyDelay(1000);

}

}