LAB 3 REPORT: EVALUATING MIXED-SIGNAL CLOSED-LOOP EMBEDDED SYSTEMS

Arpad Voros, aavoros@ncsu.edu, ECE 560

EVALUATING OPEN-LOOP CONTROL

STARTER CODE

1. **What is the switching frequency for converter? Monitor the digital signal BUCK_DRV (also called Q Drive), available on J2 at pin 13. Or examine the frequency of the ripple in I_{LED}.**

When running on the open-loop controller (with **g_duty_cycle** at 100, **PWM_PERIOD** at 300), the switching frequency is roughly 80 kHz. This is due to the **PWM_PERIOD**, since the frequency is calculated in the following way:

$$\frac{48\text{MHz}}{2 \times \text{PWM\_PERIOD}} = f_{switching}$$

Where **PWM_PERIOD** = 300, so f_switching = 80 kHz

2. **What is the control loop frequency? Monitor the digital signal Control_HBLED.**

For the open-loop controller (with **g_duty_cycle** at 100, **PWM_PERIOD** at 300), the control loop frequency is 180 kHz. This is measured by observing the Control_HBLED period from DIO 2 on the AD2 (Debug Pin 2 on the KL25Z)

3. **Complete the table below. Run the code, modify *g_duty_cycle* using a debugger variable watch window, and measure average and peak-to-peak voltages across R10 (which will determine I_{LED}), available as V_{S--} on J13. For the last row, you'll need to adjust *g_duty_cycle* until the average LED current matches the specified value.**

g_duty_cycle	Average I_{LED} from Oscilloscope	Approximate Average Value of Variable *measured_current*	Minimum I_{LED}	Maximum I_{LED}	I_{LED} Ripple Current = Maximum I_{LED} – Minimum I_{LED}
100	7 mA	(6 + 12) / 2 = 9	3.7 mA	12.7 mA	9 mA
150	14.5 mA	(10 + 23) / 2 = 16.5	8.5 mA	25 mA	16.5 mA
239	**32 mA**	(25 + 46) / 2 = 35.5	23 mA	47 mA	24 mA

All values above were calculating by using measurements from the AD2. The approximate average value of **measured_current** was made by using the debug watch window of the uVision IDE, observing the smallest and largest values, and averaging them.

A **g_duty_cycle** of 239 properly recreated a 32 mA signal on the AD2. This caused a ripple current of 24 mA. This ripple current will be used in many calculations in the following problems within this lab to offset errors.

4. **How does the variable *measured_current* compare with the average I_{LED} determined with the oscilloscope? If there is much error, what do you think causes it?**

It is noted that despite being close, the **measured_current** values do not align exactly with the current calculations made on the oscilloscope. In fact, the real current seems to be consistently lower by 2-3 mA as opposed to the approximate average of **measured_current**. This can be due to the fact that the AD2 measurements and/or DAC measurements are not ideal, and that a very rough approximation was made in calculating the average.

5. **Take a mixed-signal screenshot of I_{LED} (with average value of 32 mA) showing two cycles of its ripple and include it in your report.**

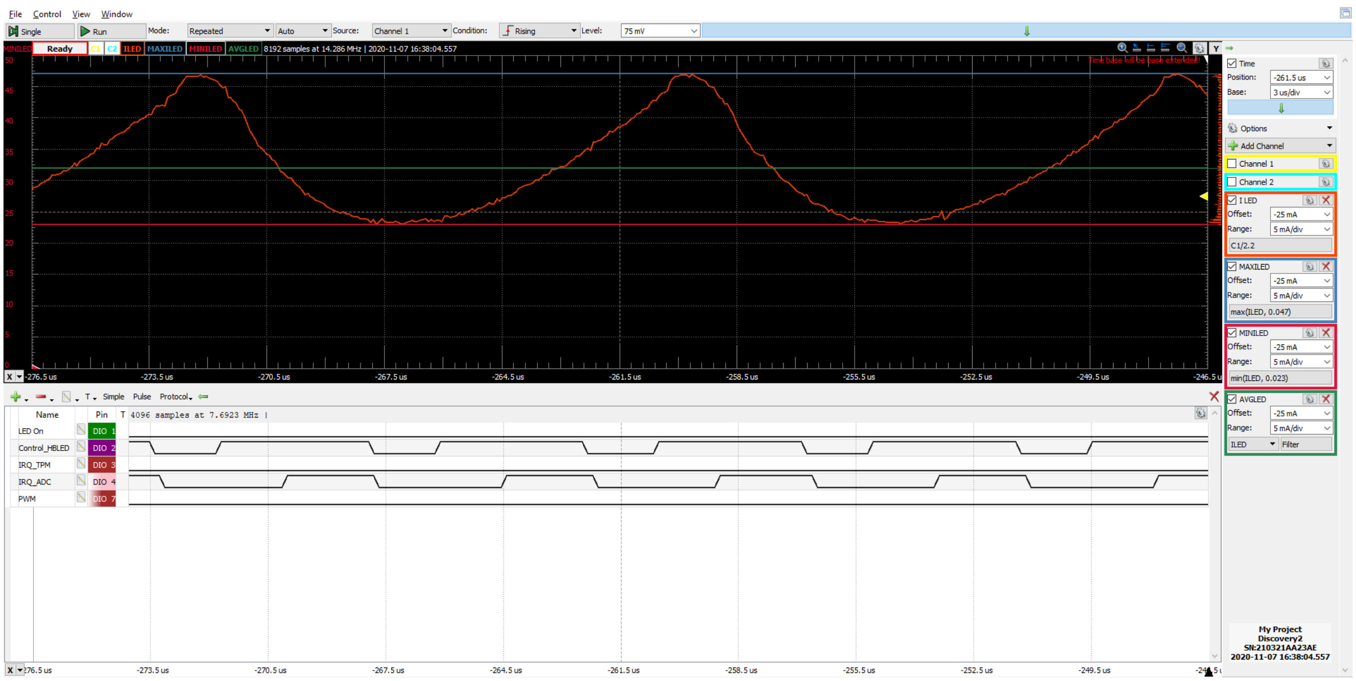


Image of the Open-Loop, No Controller, 32 mA Case

EVALUATING CLOSED-LOOP CONTROL WITHOUT TRANSIENTS

ASYNCHRONOUS SAMPLING

6. **Complete the following table. Change *g_control_mode* to select the different controllers. In the last column compute the maximum error due to the controller: (maximum I_{LED} minus minimum I_{LED}) minus I_{LED_Ripple} for the open-loop 32 mA case (from the table above).**

Controller	Control Loop Frequency	Minimum I_{LED}	Maximum I_{LED}	Peak-to-Peak I_{LED} Ripple Current from Switching	Peak-to-Peak I_{LED} Current Error from Controller
Bang-Bang (1)	158.7 kHz	2 mA	112 mA	110 mA	86 mA
Incremental (2)	160 kHz	15 mA	110 mA	95 mA	71 mA
Proportional (3)	155 kHz	20 mA	48 mA	28 mA	4 mA
Fixed Point PID (5)	59 kHz	21 mA	49 mA	28 mA	4 mA

One thing to note is the maximum current for the Incremental control-system overshot to 110 mA multiple times during the open-loop demonstration (no blinking, average current of 32 mA). On average, the ripple current was smaller, but the peak-to-peak ripple remained large due to this repeated overshoot. Some additional observations:

- Control-loop frequency for Bang-Bang, Incremental, and Proportional controllers all lie roughly around 150-160 kHz. The control-loop frequency using the PID gets drastically reduced.
- Bang-Bang and Incremental controllers have little-to-no control over the current, and both result in large ripples. However, Incremental was able to keep the ripple lower on average, whereas Bang-Bang was not.
- Proportional and PID controllers were able to drastically reduce the ripple current, to where there is only an error of 4 mA for each (28 mA – 24 mA (from the first table))

7. **Take a mixed-signal screenshot of about 20 cycles of I_{LED} for any one control mode (your choice of which) and include it in your report.**

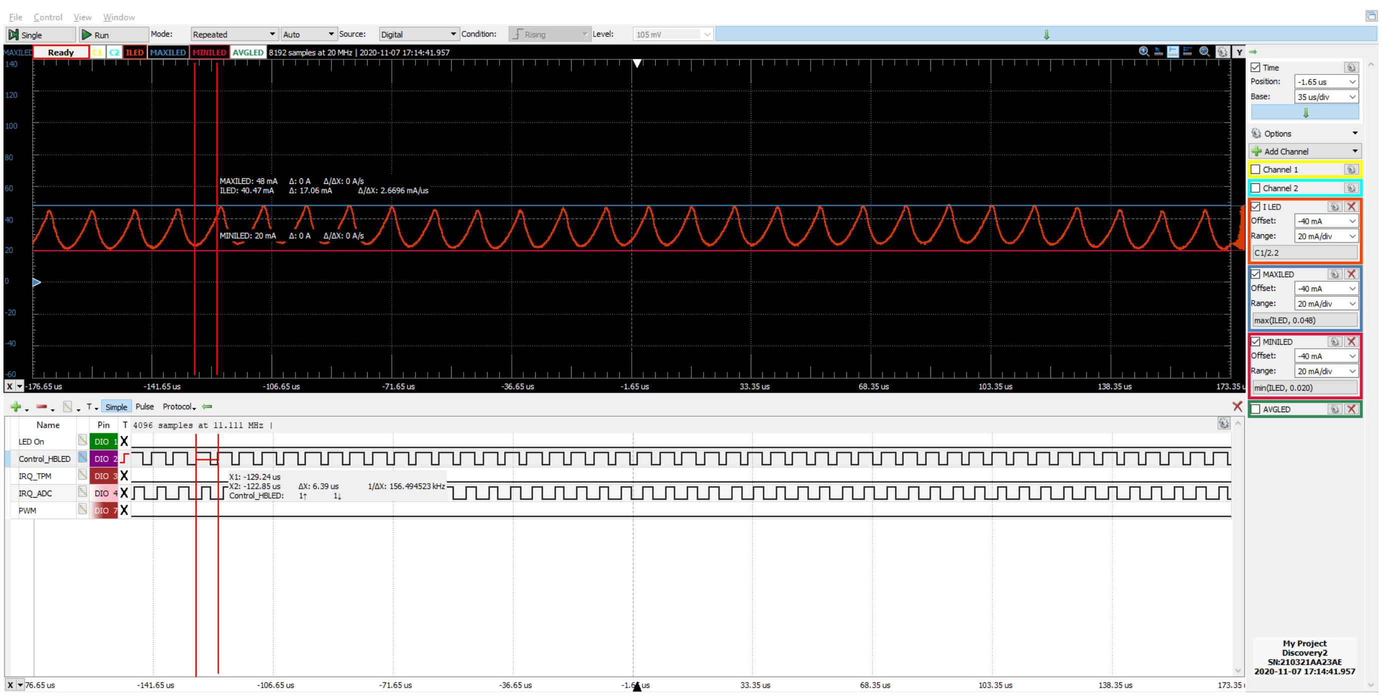


Image of the Proportional Controller, Open-Loop 32 mA Case

One can observe the error ripple of 4 mA (where the wave shifts slightly up and down), and the overall ripple being 28 mA, and the control-loop frequency being 155 kHz

SYNCHRONOUS SAMPLING

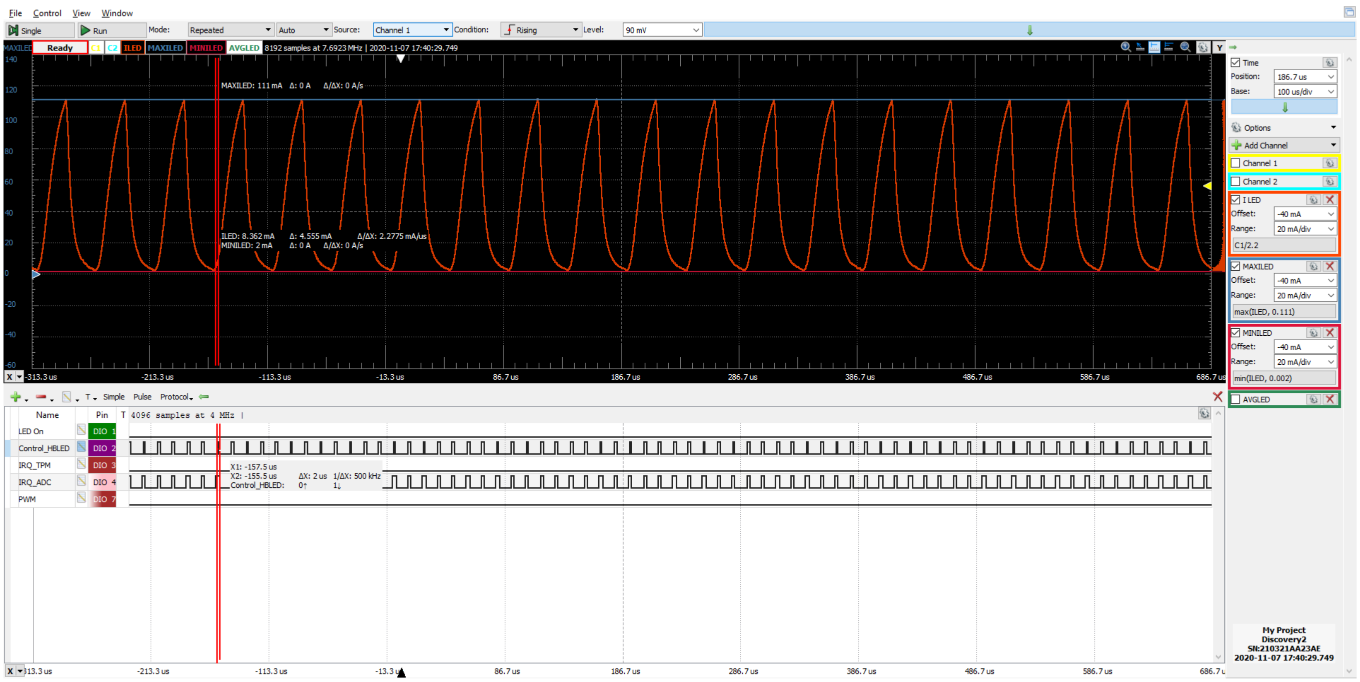
8. **Complete the following table. Change *g_control_mode* to select the different controllers. Compute the maximum error due to the controller: (maximum I_{LED} minus minimum I_{LED}) minus I_{LED_Ripple}. Use the logic analyzer window to determine the duration of Control_HBLED.**

	Minimum I_{LED}	Maximum I_{LED}	Peak-to-Peak I_{LED} Ripple Current from Switching	Peak-to-Peak I_{LED} Current Error from Controller	Control_HBLED Duration
Bang-Bang	2 mA	111 mA	109 mA	85 mA	2.06 µs
Incremental	17 mA	45 mA	28 mA	4 mA	2.06 µs

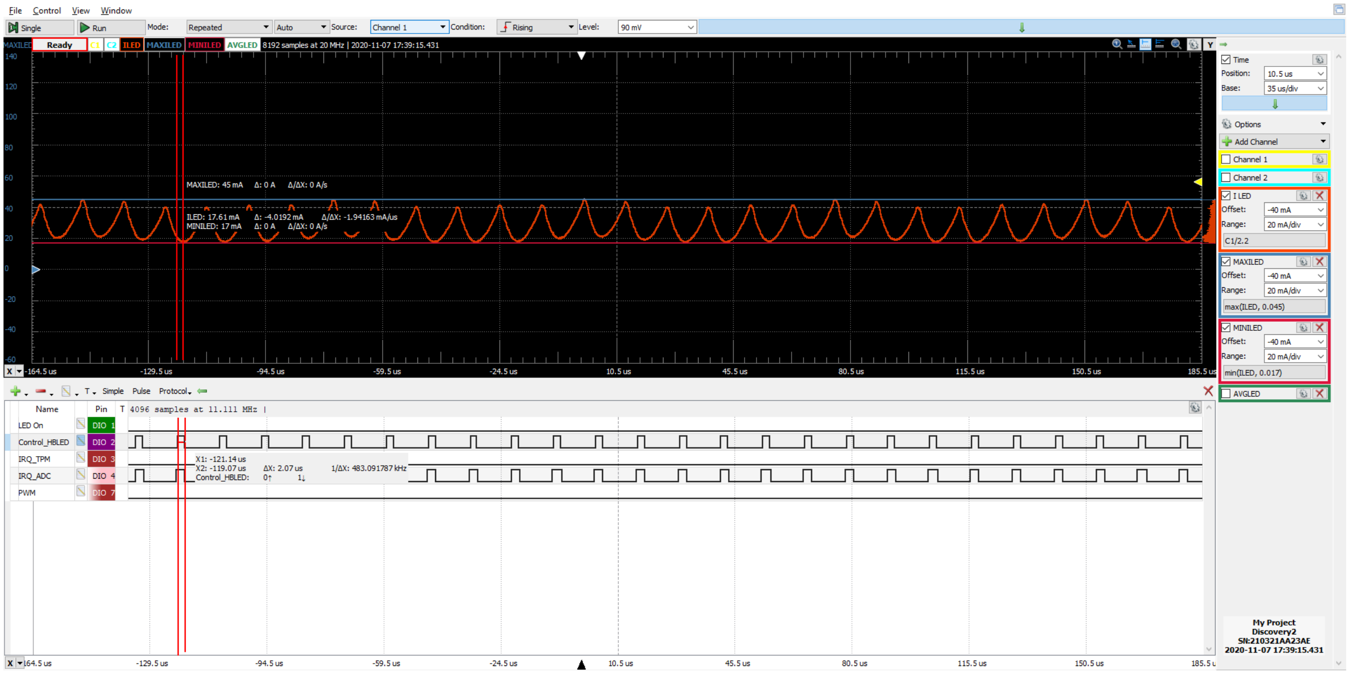
Proportional	18 mA	42 mA	24 mA	0 mA	2.15 µs
Fixed Point PID	18 mA	42 mA	24 mA	0 mA	7.31 µs

When adjusting to synchronous sampling, the Incremental controller's peak-to-peak current drastically improves. There is no overshoot anymore. The other controllers all have relatively the same minimum/maximum current values as before (with slight improvements), but we can see that the Proportional and PID controllers now have a 0 mA error, where all the I_{pp} values come directly from the open-loop 32 mA case. The duration of Control_HBLED is observed, which is related to the control-loop frequencies found above.

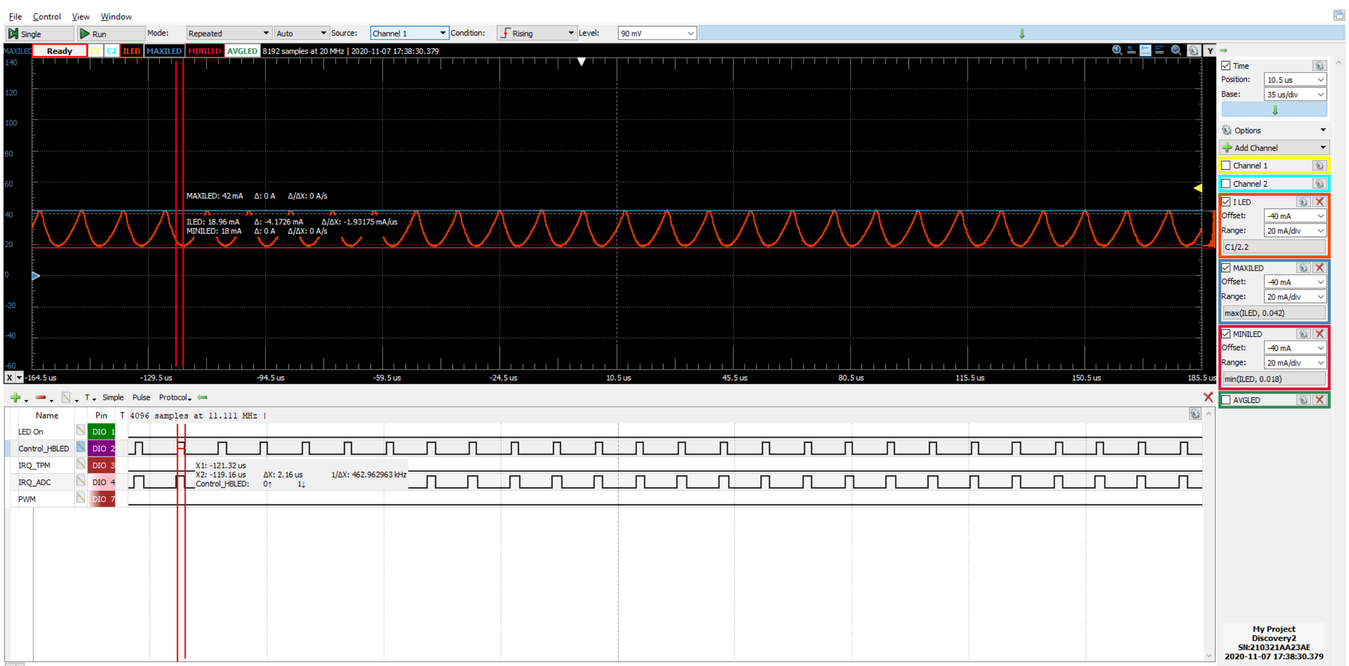
9. **Take mixed-signal screenshots of about 20 cycles of I_{LED} for each of the control methods and include them in your report.**



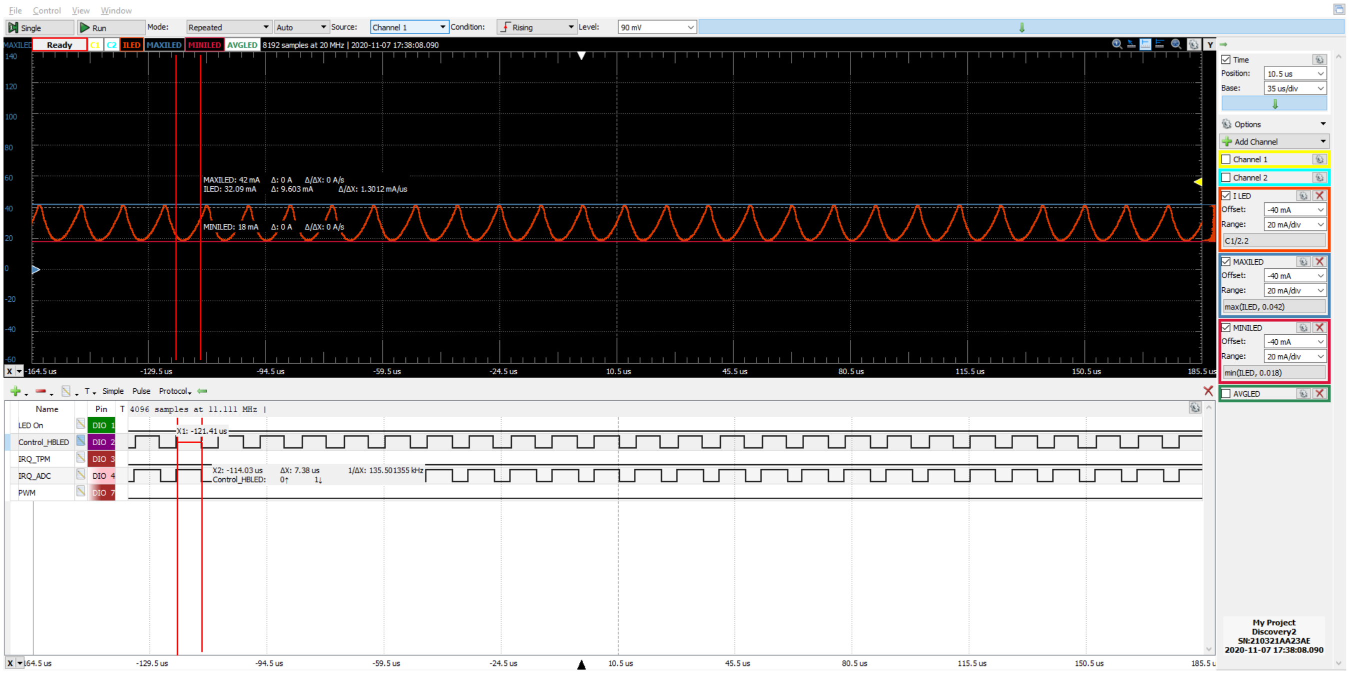
*Bang-Bang Controller, I_{pp} & **CONTROL_HBLED** duration shown*



*Incremental Controller, I_{pp} & **CONTROL_HBLED** duration shown*



*Proportional Controller, I_{pp} & **CONTROL_HBLED** duration shown*

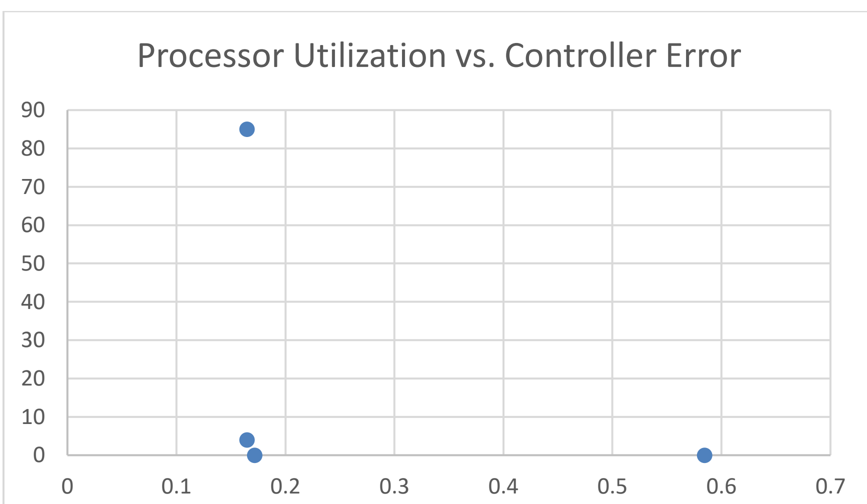


*PID Controller, I_{pp} & **CONTROL_HBLED** duration shown*

All images use the same scale, thus can be proportionally compared to one another.

10. **Create one scatter plot showing the controller error (mA, vertical) vs. processor utilization (%, horizontal) for the control approaches. Calculate processor utilization as Control_HBLED duration * control loop frequency.**

Regardless of controller, control-loop frequency was 80 kHz due to synchronization. Since in my case, both the Proportional and PID controllers had 0 mA error, the scatter plot does not seem to show much. However, I am assuming that the message trying to be conveyed was that there is a trade-off between processor utilization and low error. Increase time calculating the current (increased processor time), there should be less error.



EVALUATING CLOSED-LOOP CONTROL WITH TRANSIENTS

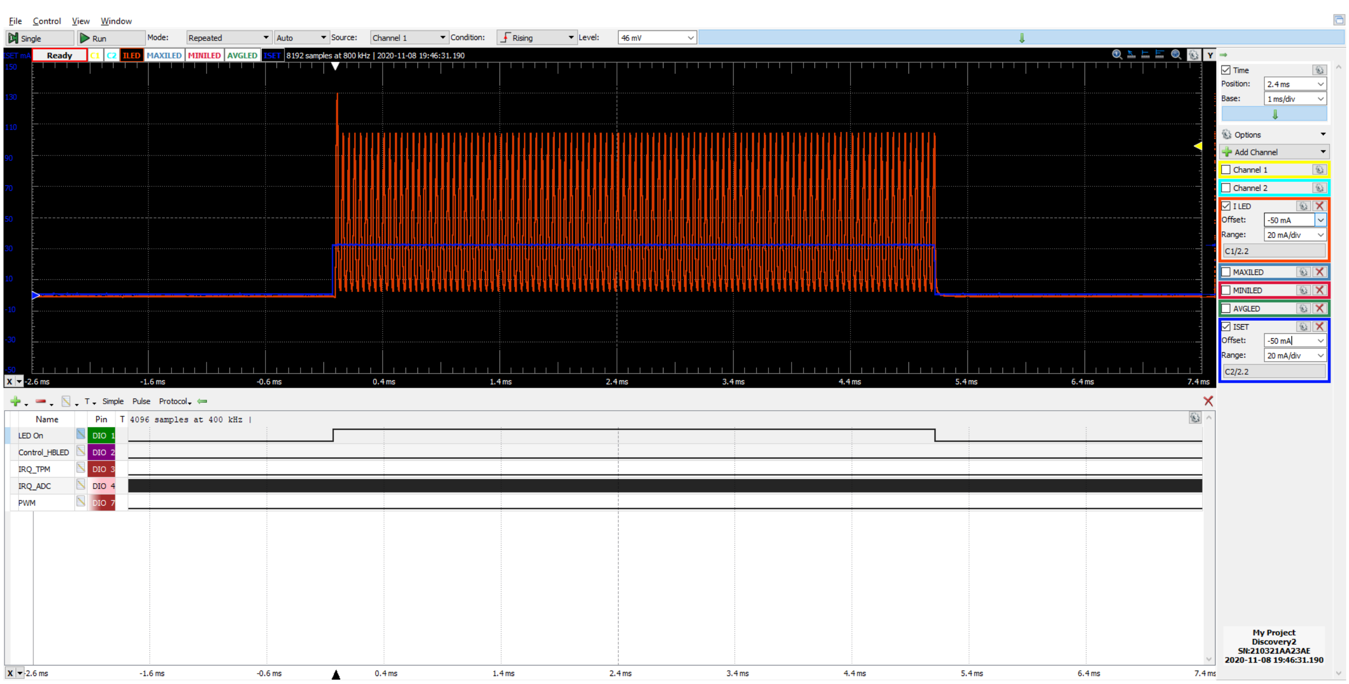
11. Complete the following table. Change *g_control_mode* to select the different controllers. In the last column compute the maximum error due to the controller: (maximum I_{LED} minus minimum I_{LED}) minus the peak-to-peak I_{LED} ripple for the corresponding open-loop case (as you determined previously).

	Delay From $I_{setpoint}$ Change Until I_{LED} First Reaches $I_{setpoint}$		I_{LED_Max} (includes overshoot)	Peak-to-Peak I_{LED} Ripple Current from Switching		Peak-to-Peak I_{LED} Current Error from Controller	
	0 to 32 mA	32 to 0 mA	32 mA	32 mA	0 mA	32 mA	0 mA
Bang-Bang	29 µs	19 µs	133 mA	104 mA	3.7 mA	80 mA	0.2 mA
Incremental	335 µs	360 µs	43 mA	25 mA	3.6 mA	1 mA	0.1 mA
Proportional	596 µs	912 µs	37 mA	21.7 mA	7.7 mA	0 mA	4.1 mA
Fixed Point PID	211 µs	478 µs	41.5 mA	22.6 mA	3.5 mA	0 mA	0 mA

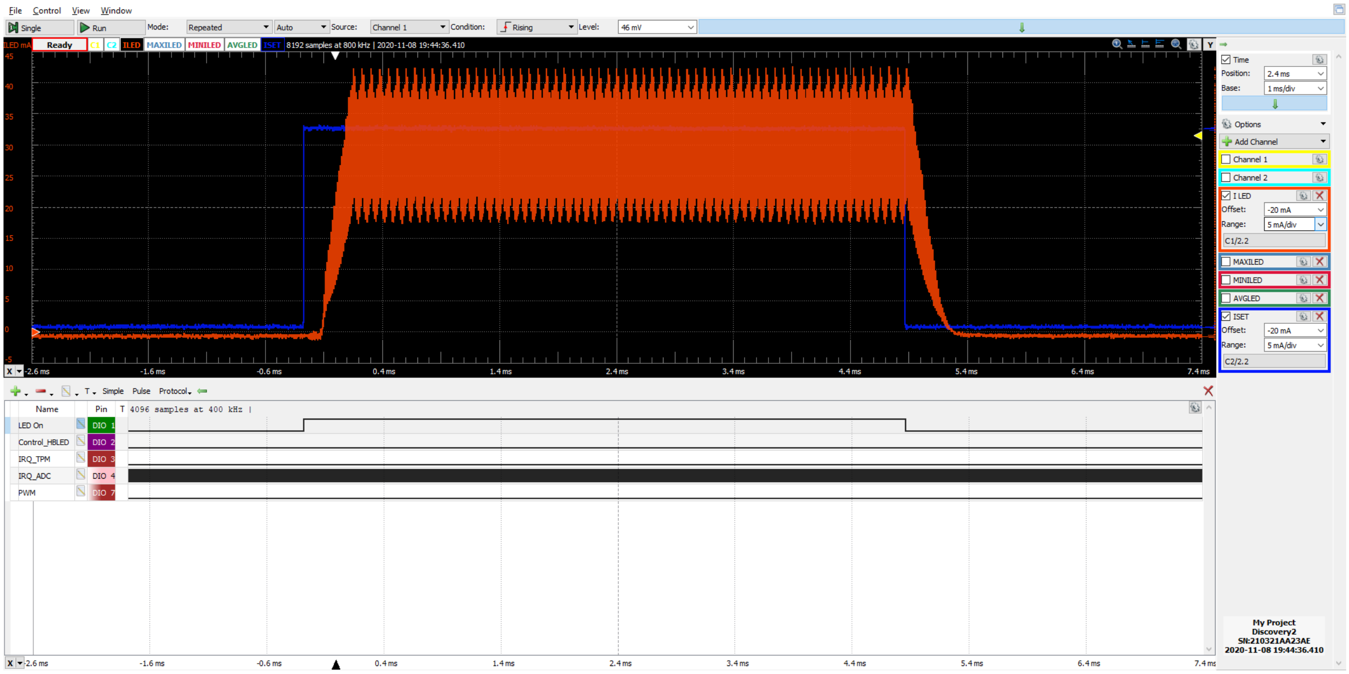
When observing the transient blinks, it is immediately observed how differently all these controllers behave. Bang-Bang is pretty self-explanatory, it is explosive so there is a short rise/fall time, but large overshoot and ripple current. The incremental controller increases and decreases at the same rate and has little-to-no overshoot and error. The Proportional controller has a slow rise/fall time, but there is even less overshoot and ripple current error. Similar to the Proportional controller, the PID behaves the same way but with improvements on the rise/fall times. Images for each moment of transience for every controller are shown below.

In addition, the Proportional and PID controller both seemed to have a lower I_{pp} than the normal ripple error of 24 mA. This is observed again with the Proportional controller in future sections.

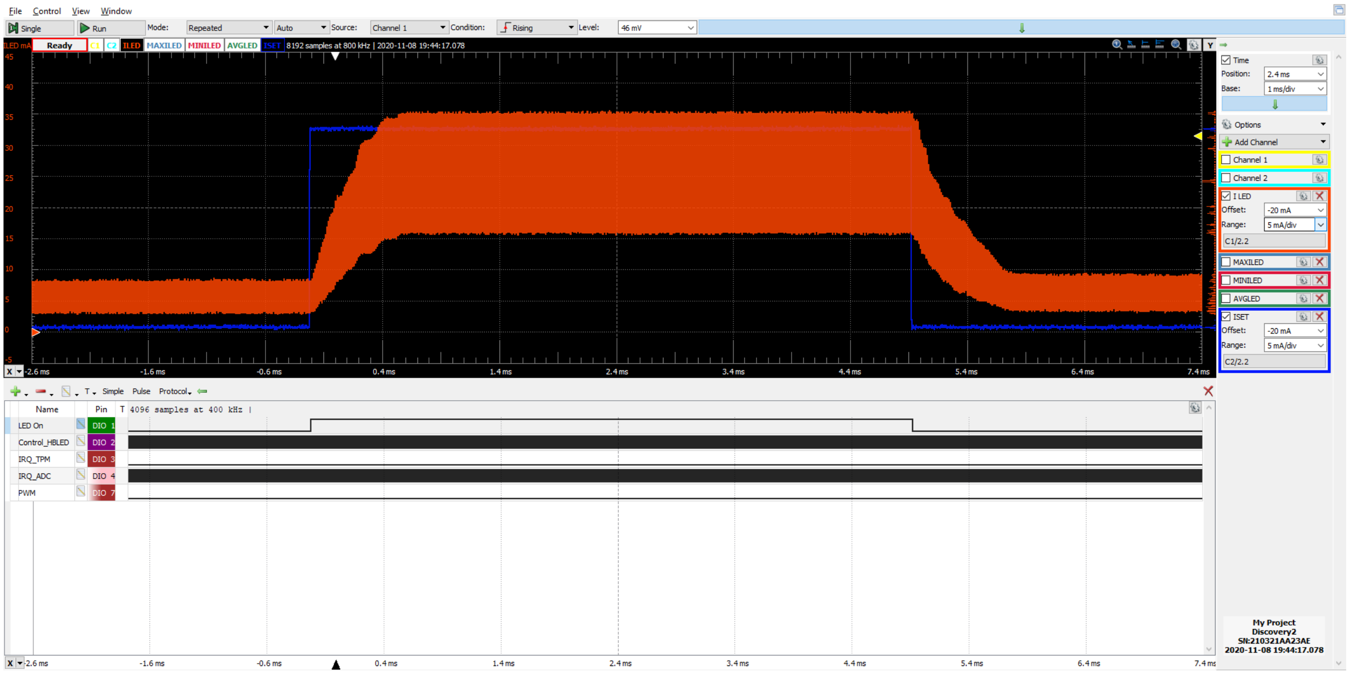
12. Take mixed-signal screenshots of I_{LED} showing one flash (including some of the 0 mA times before and after) for each of the control methods and include them in your report.



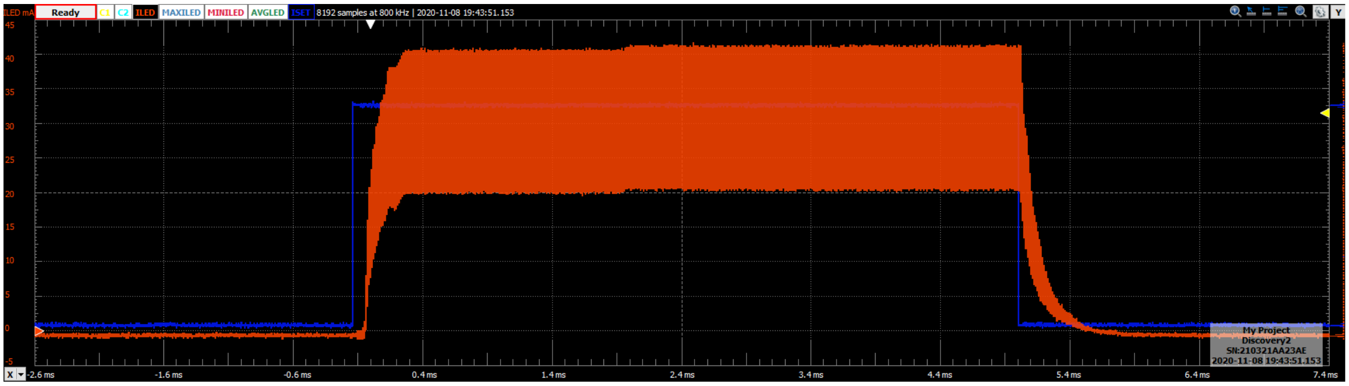
Bang-Bang Controller, showing period of LED on



Incremental Controller, showing period of LED on



Proportional Controller, showing period of LED on



PID Controller, showing period of LED on

Incremental, Proportional, and PID controllers all use the same scope scale. Bang-Bang controller had to be significantly zoomed-out due to it's large variation in ripple current as well as large initial overshoot.

(PID image accidentally cut-off the rest of the window, only showing analog scope)

ECE 560 ONLY: IMPROVING CONTROL SYSTEM PERFORMANCE

RAISING SWITCHING AND CONTROL LOOP FREQUENCY

OPEN-LOOP AND CLOSED-LOOP WITHOUT TRANSIENTS:

13. What is the new switching and control frequency? What is the open-loop ripple current for I_{LED} = 32 mA at this frequency? How does well does each controller work at its new frequency without transients (no HBLED flashing)? Complete the following table. Change *g_control_mode* to select the different controllers. Compute the maximum error due to the controller: (maximum I_{LED} minus minimum I_{LED}) minus the peak-to-peak I_{LED} ripple for the open-loop 32 mA case.

	Maximum $f_{control}$ and $f_{switching}$	Minimum PWM_PERIOD	Open-Loop Ripple Current	Controller Error for I_{LED}
Bang-Bang	275 kHz	87	19.8 mA	15.5 mA
Incremental	275 kHz	87	19.8 mA	15.5 mA
Proportional	266 kHz	90	8 mA	8.9 mA
Fixed Point PID	109 kHz	220	31.5 mA	20.7 mA

Each of the lowest **PWM_PERIOD** values were found by continuously decreasing **PWM_PERIOD** and ensuring the KL25Z's LED responded properly in the following ways:

- Properly idled for a slight amount of time between **Control_HBLED** and **IRQ_ADC** high moments
- Blinked when **g_enable_flash** was set to 1 and **g_control_mode** was one of the controllers show above
- Constantly drew current when **g_enable_flash** was 0, **g_set_current** was 32, and **g_control_mode** was **OpenLoop**

Bang-Bang and Incremental ended up using the same **PWM_PERIOD**, so all the data points are shared

CLOSED-LOOP WITH TRANSIENTS

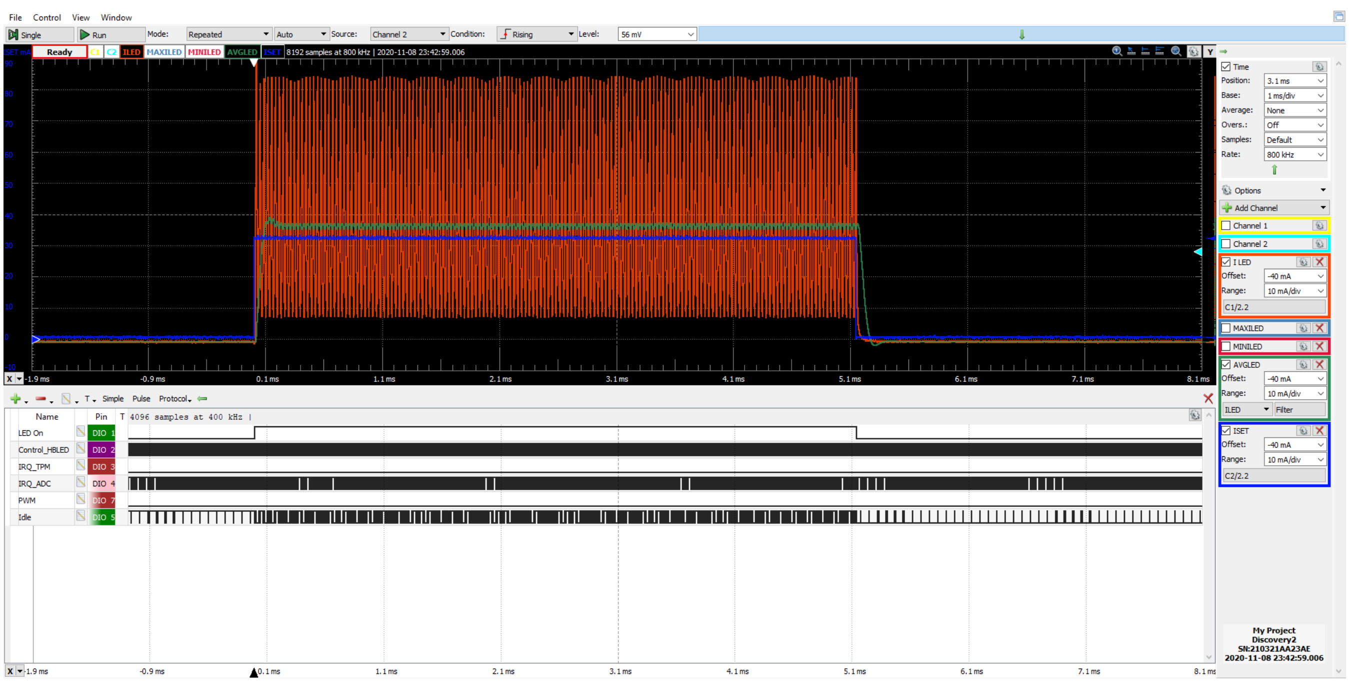
14. Complete the following table. Change *g_control_mode* to select the different controllers. In the last column compute the maximum error due to the controller: (maximum I_{LED} minus minimum I_{LED}) minus the peak-to-peak I_{LED} ripple for the corresponding open-loop case (as you determined previously).

	Delay From $I_{setpoint}$ Change Until I_{LED} First Reaches $I_{setpoint}$		I_{LED_Max} (includes overshoot)	I_{LED_Ripple}		$I_{Steady\ State\ Controller\ Error}$	
	0 to 32 mA	32 to 0 mA	32 mA	32 mA	0 mA	32 mA	0 mA
Bang-Bang	15.5 µs	30 µs	76.6 mA	79.8 mA	3.7 mA	55.8 mA	0.2 mA
Incremental	147 µs	89 µs	79.7 mA	71.9 mA	3.7 mA	47.9 mA	0.2 mA
Proportional	44 µs	46 µs	37.3 mA	6.4 mA	5.2 mA	0 mA	1.7 mA
Fixed Point PID	220 µs	365 µs	51.6 mA	33 mA	3.5 mA	9 mA	0 mA

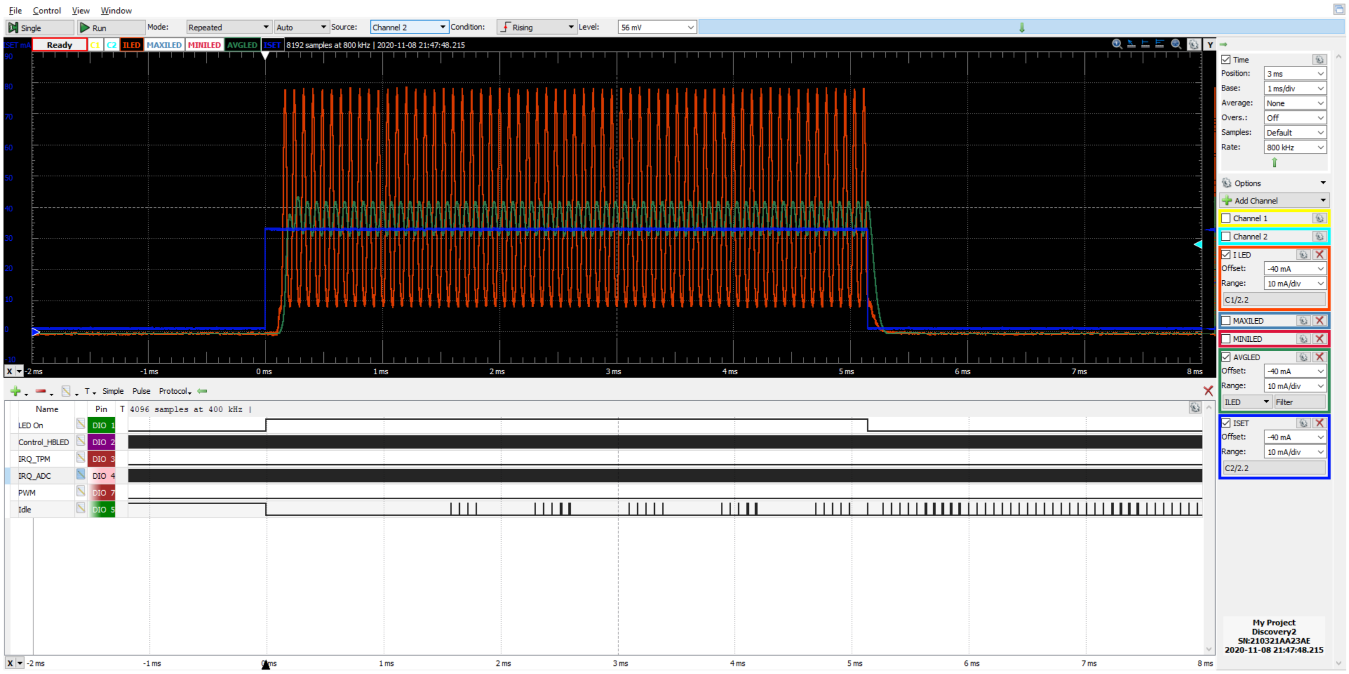
It is interesting to see the way the Proportional controller behave. The current ripple was miniscule, but the tradeoff was that there was still current being drawn when **g_set_current** was 0. The LED would be lit with blinks going off. Whereas with all the other controllers, there is a significantly larger ripple, but the LED would properly blink from off to on. This is why the rise/fall time for the Proportional controller has decreased and the tiny current ripple. The images below show what I mean.

The Proportional and PID controllers had to have their gains adjusted in **HBLED.h** file to accommodate for the change in **PWM_PERIOD** and frequency.

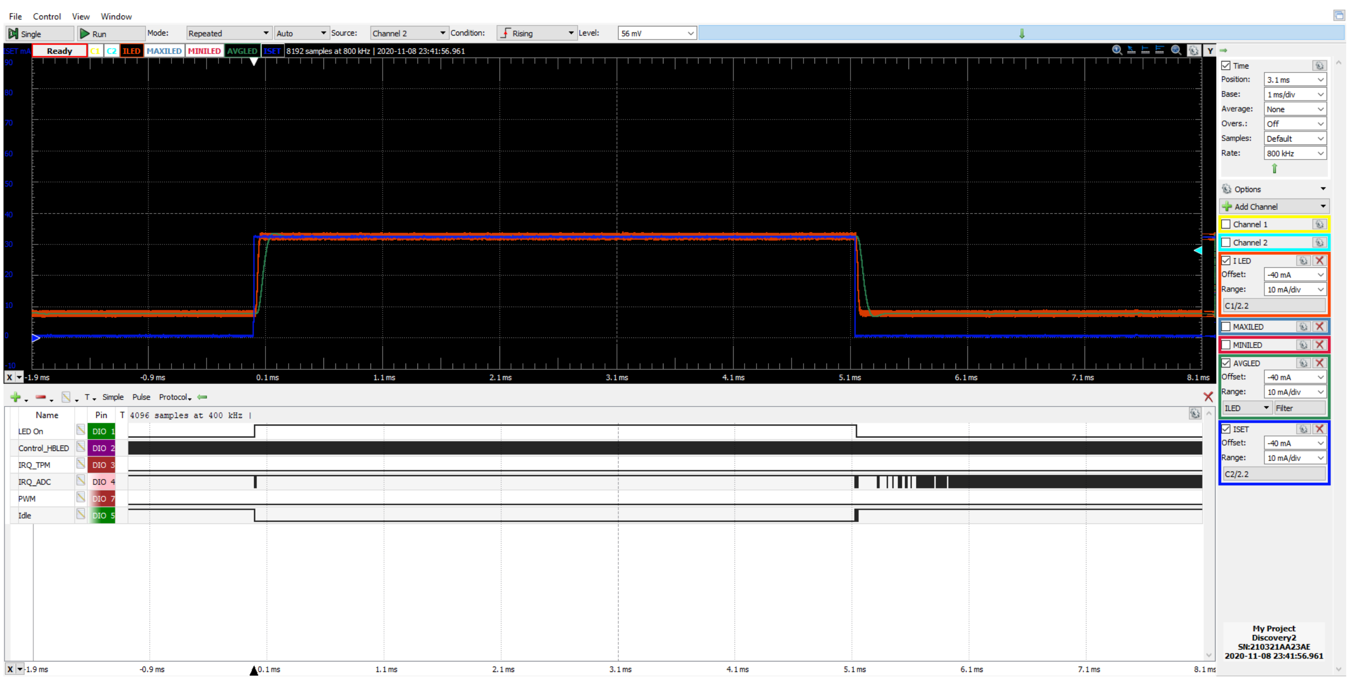
15. Take screenshots of I_{LED} showing one flash for each of the control methods and include them in your report.



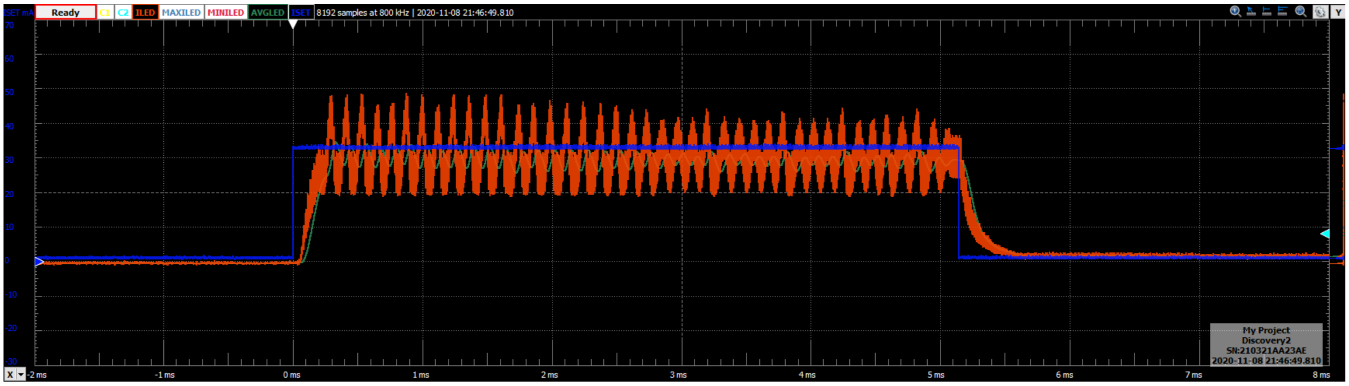
*Bang-Bang controller with **PWM_PERIOD** = 87*

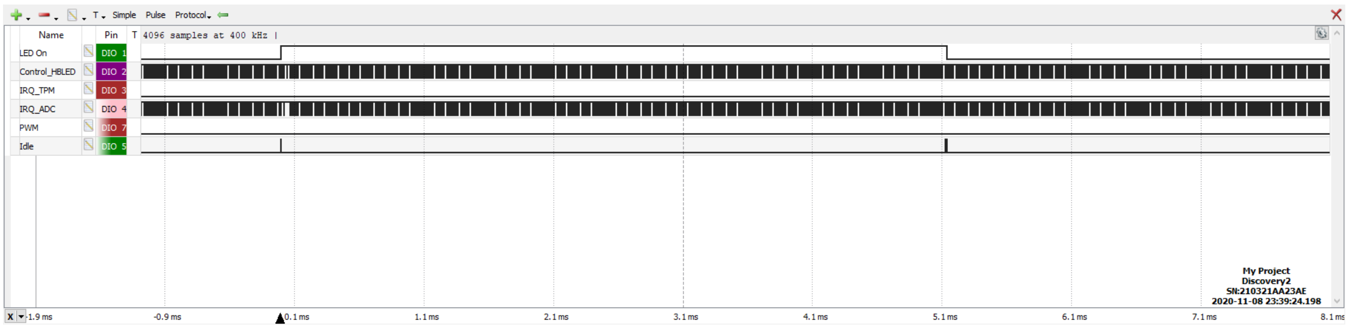


*Incremental controller with **PWM_PERIOD** = 87*



*Proportional controller with **PWM_PERIOD** = 90*





*PID controller with **PWM_PERIOD** = 220*

Proportional and PID controllers use the same scale, whereas Bang-Bang and Incremental controllers are both zoomed-out a bit. It can be observed that the Proportional controller does a magnificent job with reducing the ripple error, but it remains on during times when it should be off. The ripple and overshoot for both Bang-Bang and incremental are significantly larger than for Proportional and PID controllers.

(Again, I don't know why PID controller was unable to export both analog as well as digital images at once, but this time I caught the mistake and exported them both individually. Timestamps do show different times, because I preferred the earlier picture for the scope more)