Vinzenz Felder   Linux_Embedded: ECE497
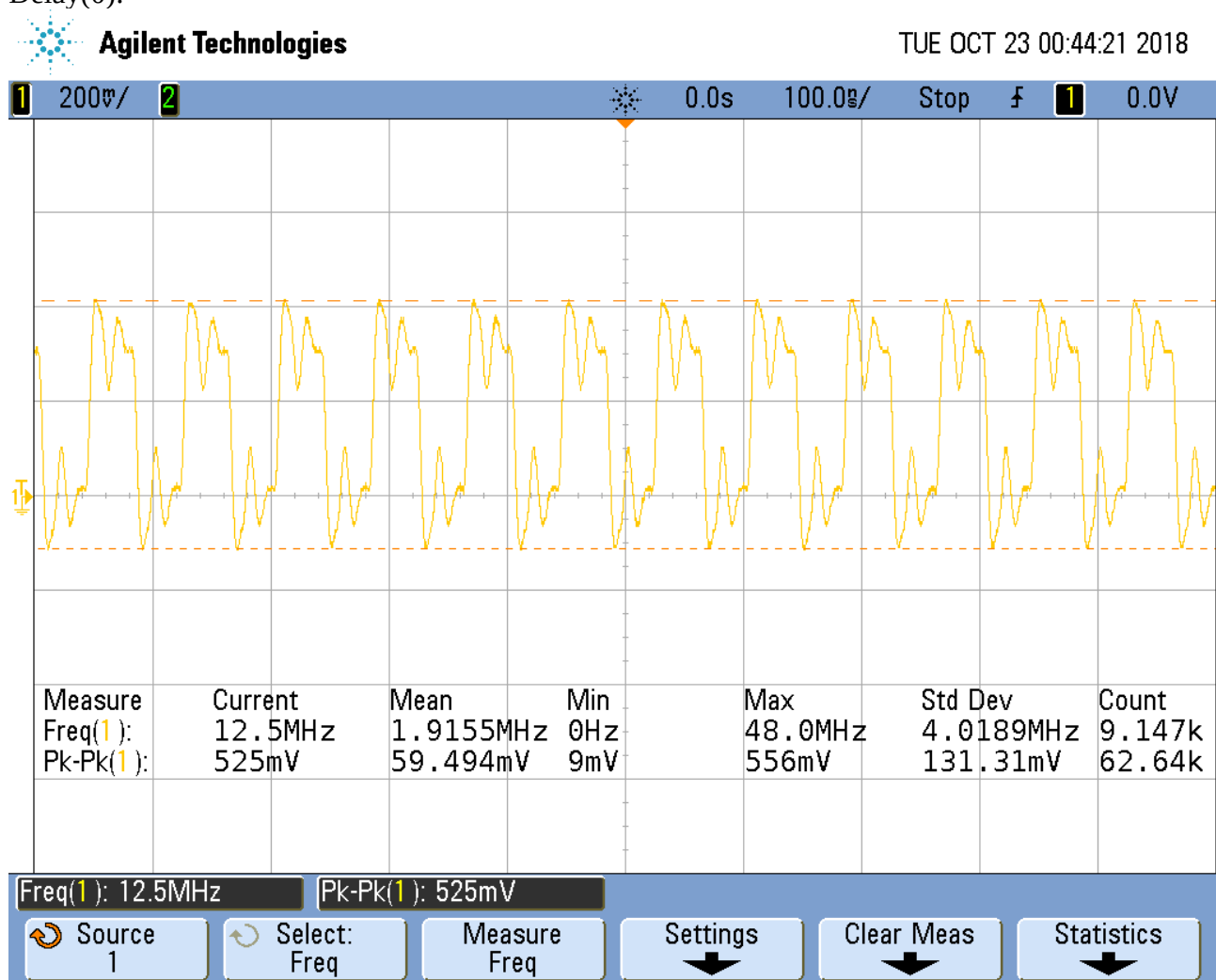
Measurement of hw09:

2.6 Blinking an LED:

Show several figures with different delays:
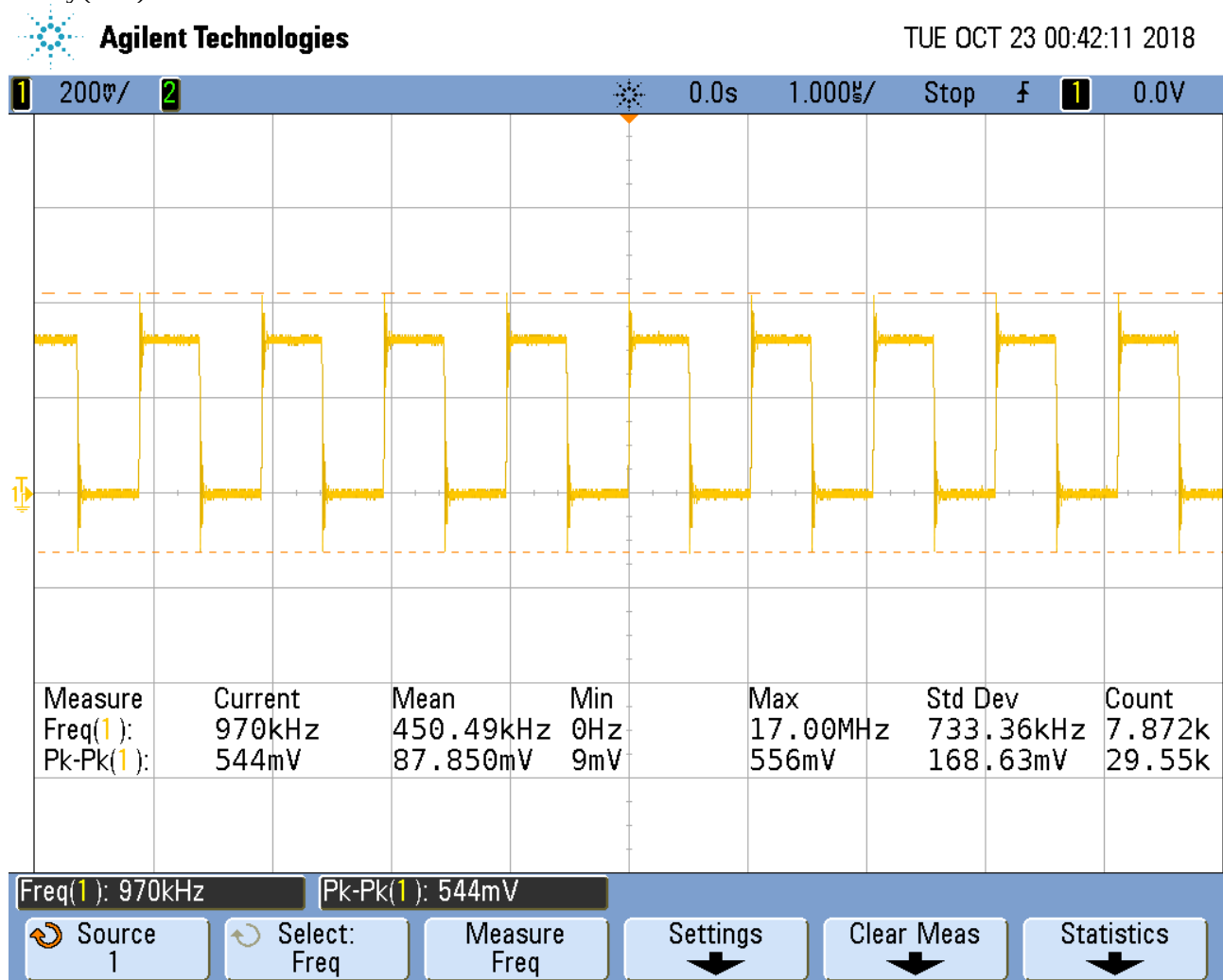
Max toggle speed: 12.5MHz
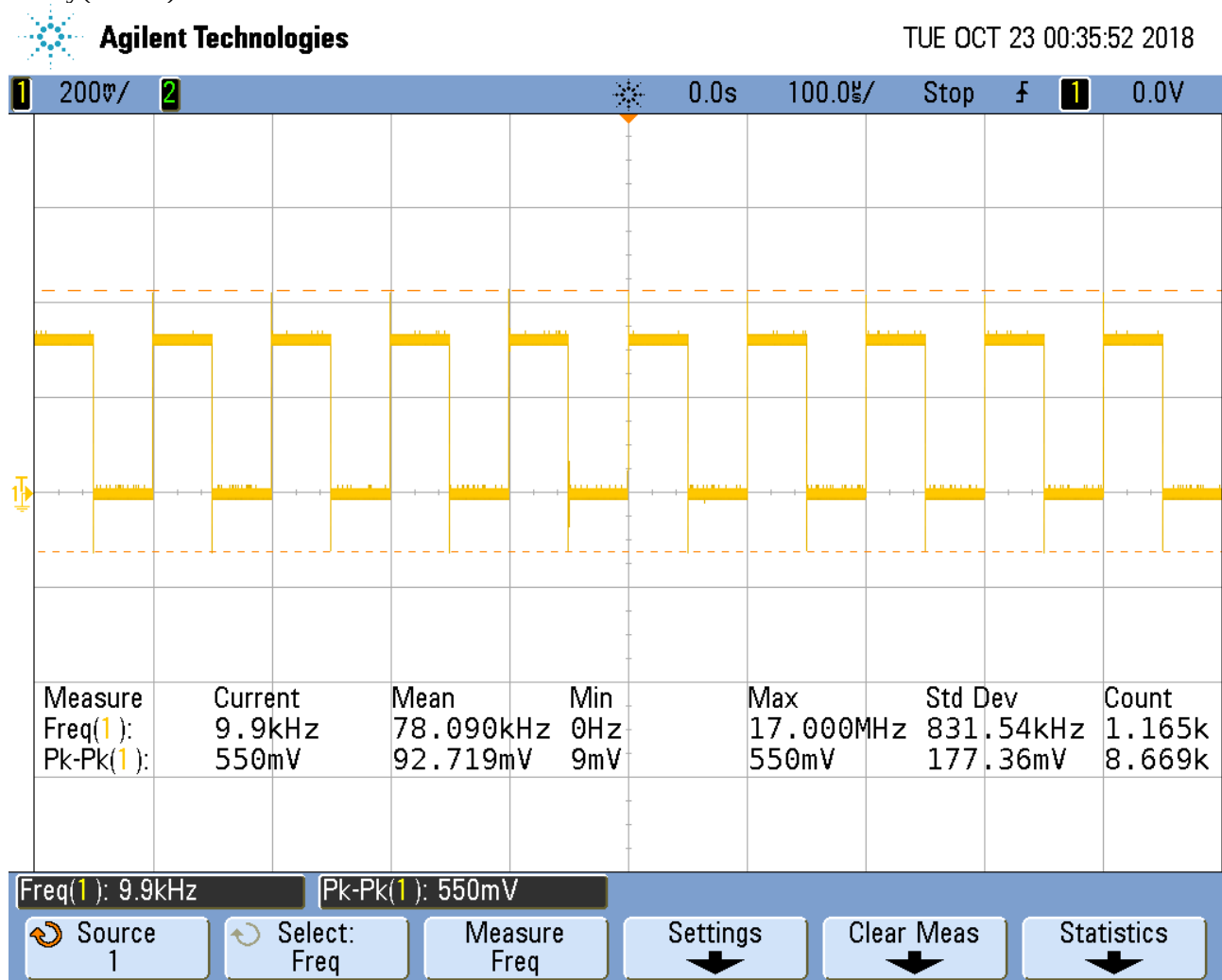
Jitter is there ++

Stability is quit good

Delay(0):

Vinzenz Felder   Linux_Embedded: ECE497

Delay(100):

Measure	Current	Mean	Min	Max	Std Dev	Count
Freq(1):	970kHz	450.49kHz	0Hz	17.00MHz	733.36kHz	7.872k
Pk-Pk(1):	544mV	87.850mV	9mV	556mV	168.63mV	29.55k

See jitter peaks?

Vinzenz Felder   Linux_Embedded: ECE497

Delay(10000):

Measure	Current	Mean	Min		Max	Std Dev	Count
Freq(1):	9.9kHz	78.090kHz	0Hz		17.000MHz	831.54kHz	1.165k
Pk-Pk(1):	550mV	92.719mV	9mV		550mV	177.36mV	8.669k

Freq(1): 9.9kHz   Pk-Pk(1): 550mV

| Source 1 | Select: Freq | Measure Freq | Settings | Clear Meas | Statistics |

5.3 PWM Generator:

I ended up 40MHz for delay(1), maximum what I could reach
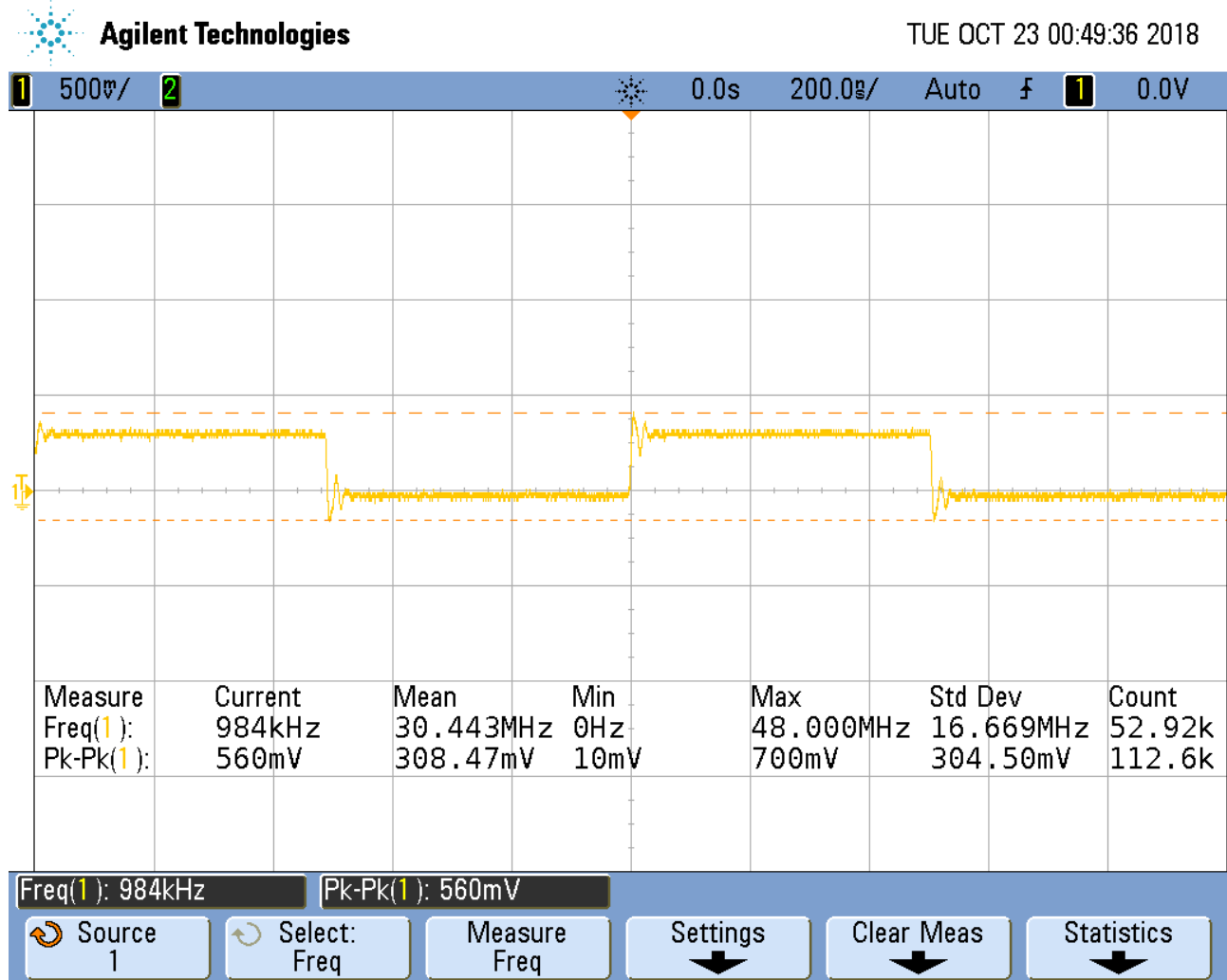Stability: Constantly changes between 38,9MHz and 42Hz, but the average is 40MHz
Standart Deviation.: 18.955MHz.
Jitter: Look Jitter in extra Figure, is there
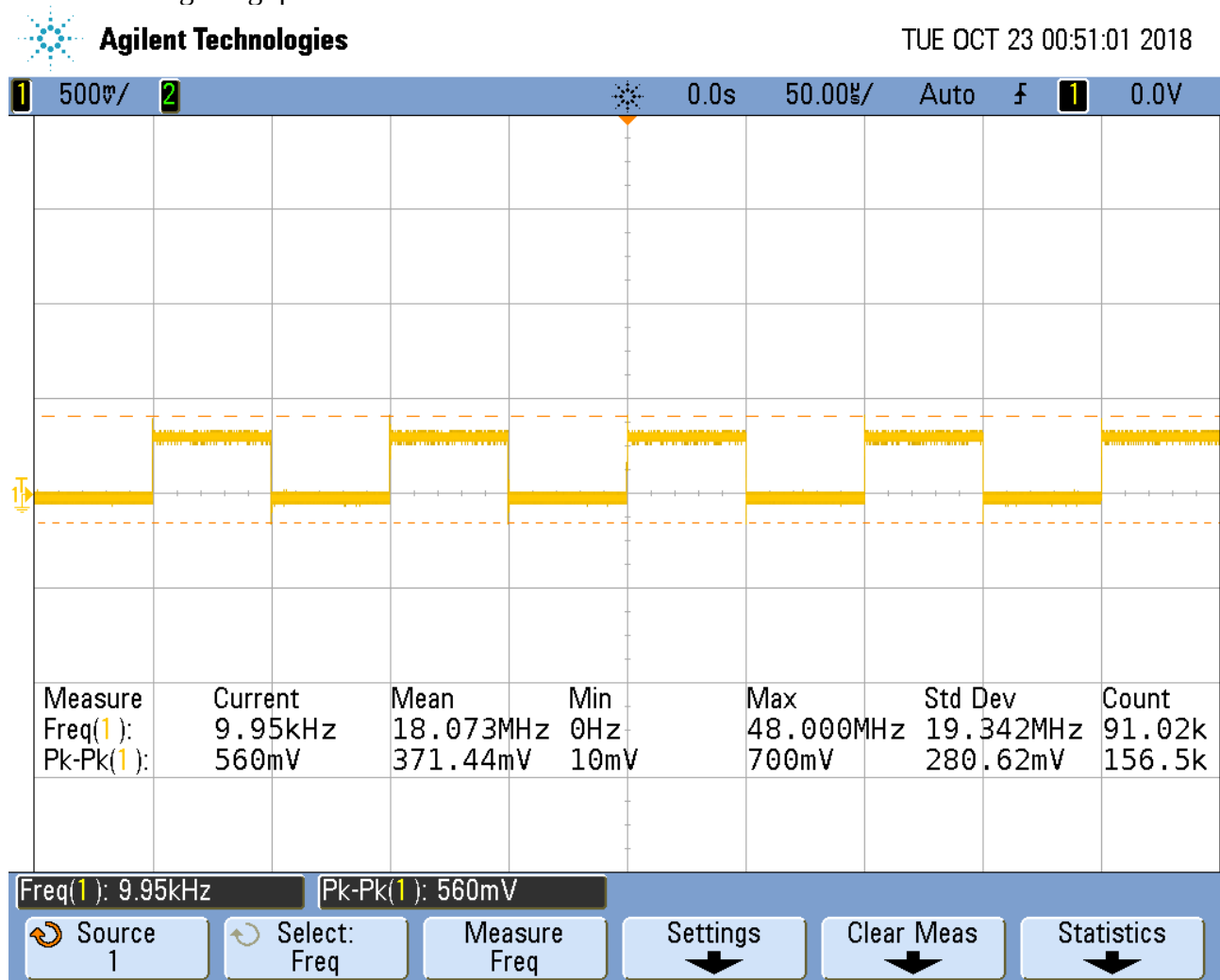Maximum speed: 40MHz
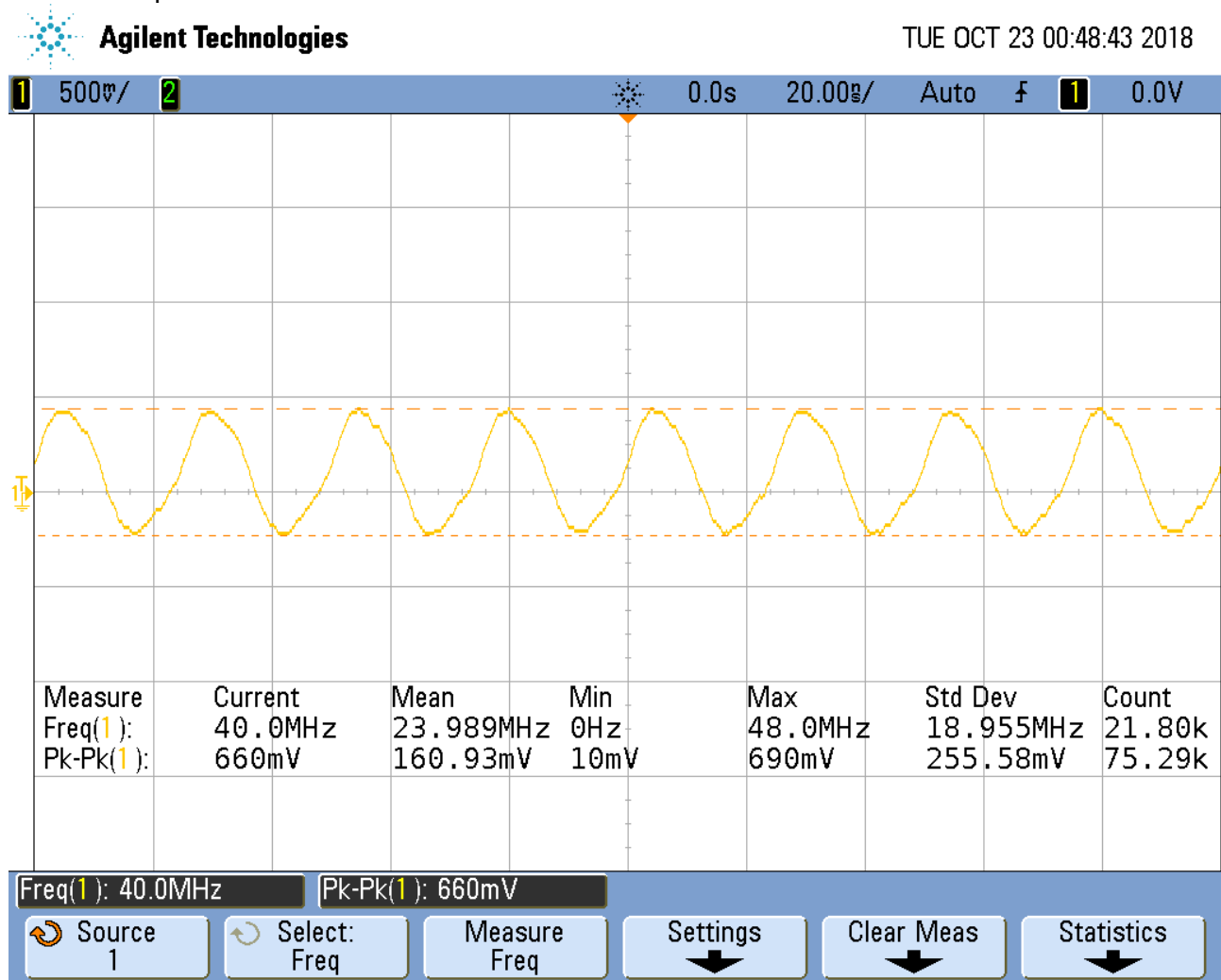
delay (100):

Compared to the previous figure the frequency is better than with the normal GPIO
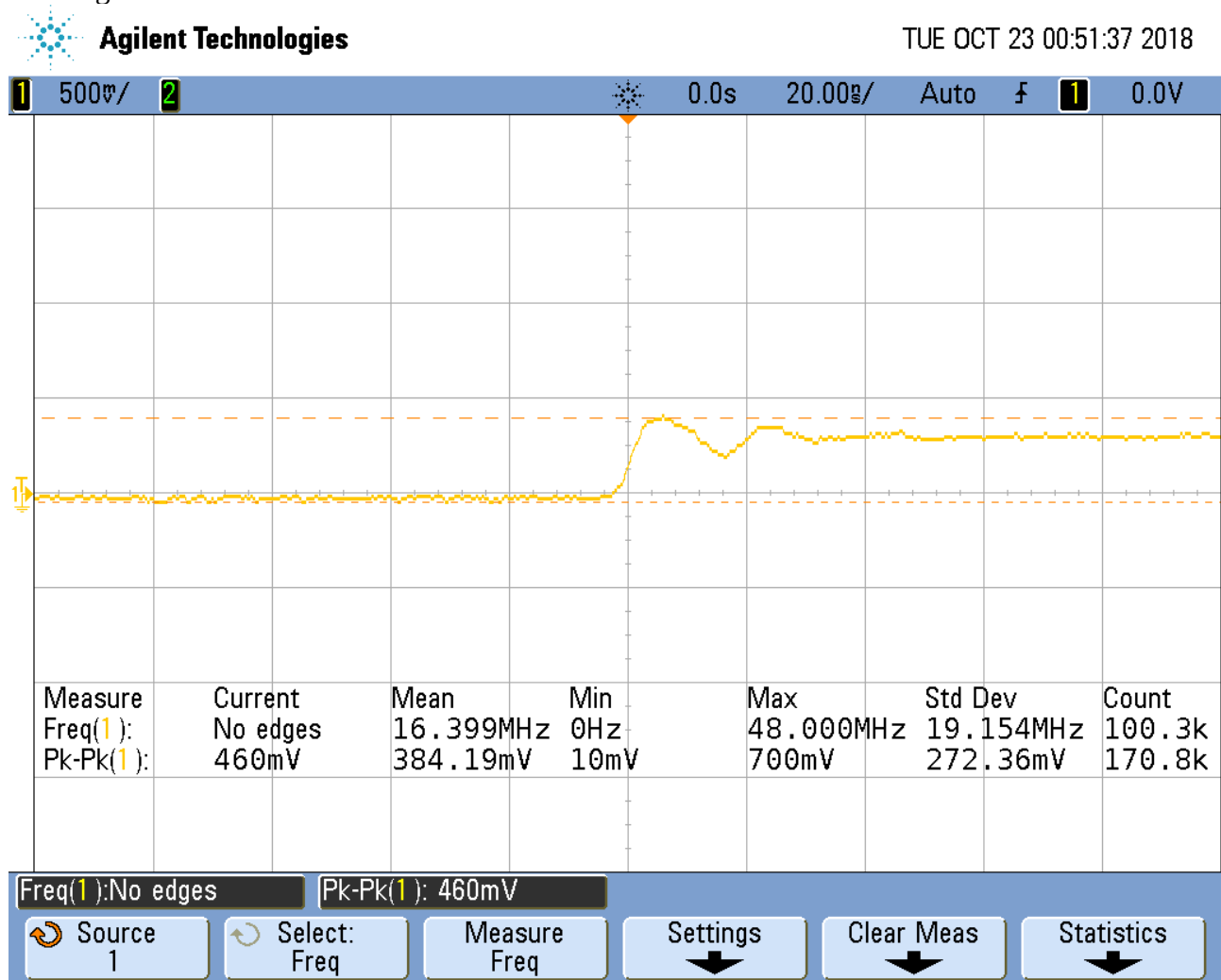
delay (10000):
Note: Jitter is getting quit better

frequency compared to the figure from blinking Led, is better. But the diffrence decreased.
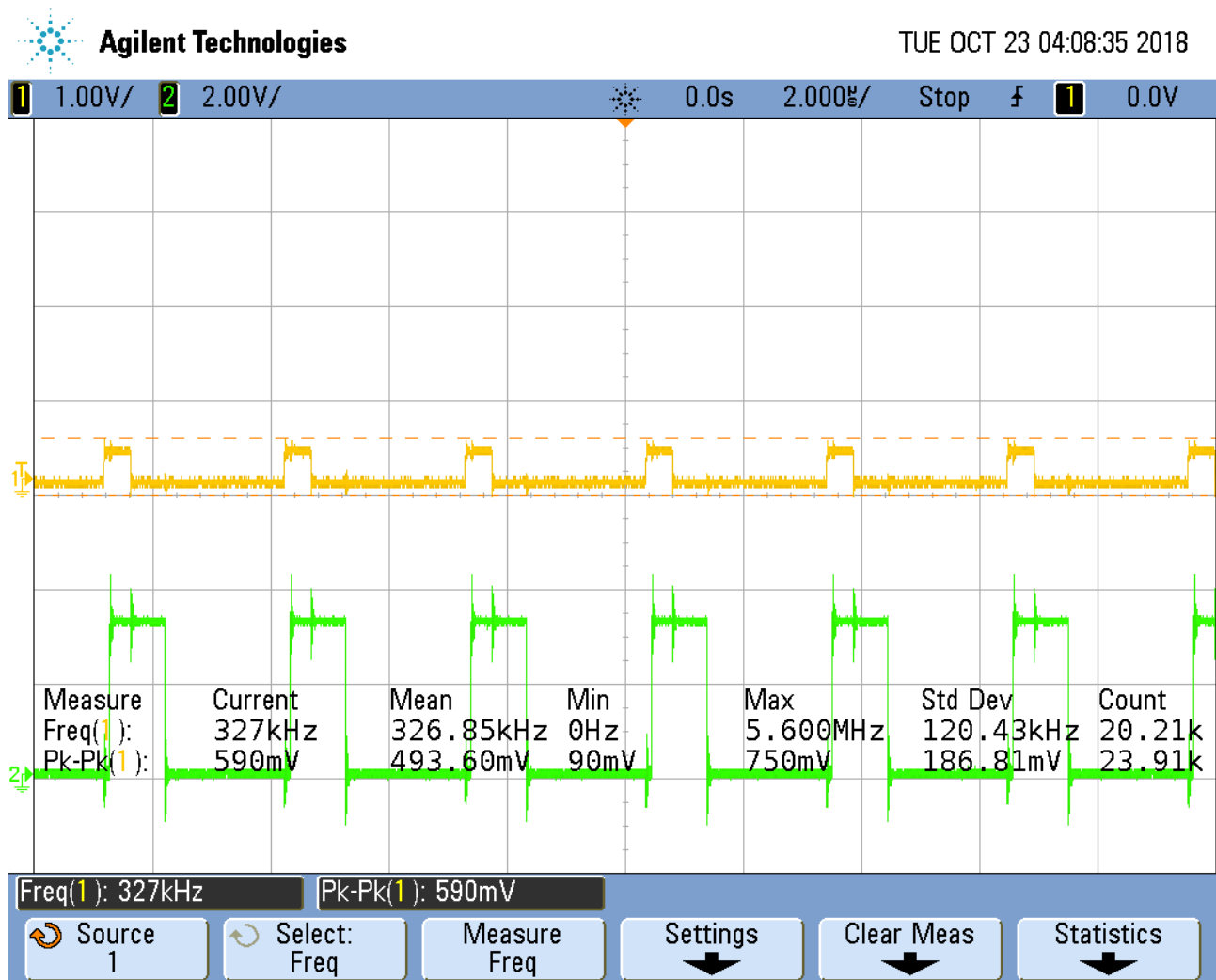
extra figure:

Jitter is shown in figure. But in my opinion still not this high than with normal GPIO output.

5.4 Controlling the PWM Frequency:

For this part i hadn't the opportunity to display all pins in one graph. Therefore I made two screenshots. First is with the pins P8_43 and P8_44, second with the following P8_45 and P8_46.
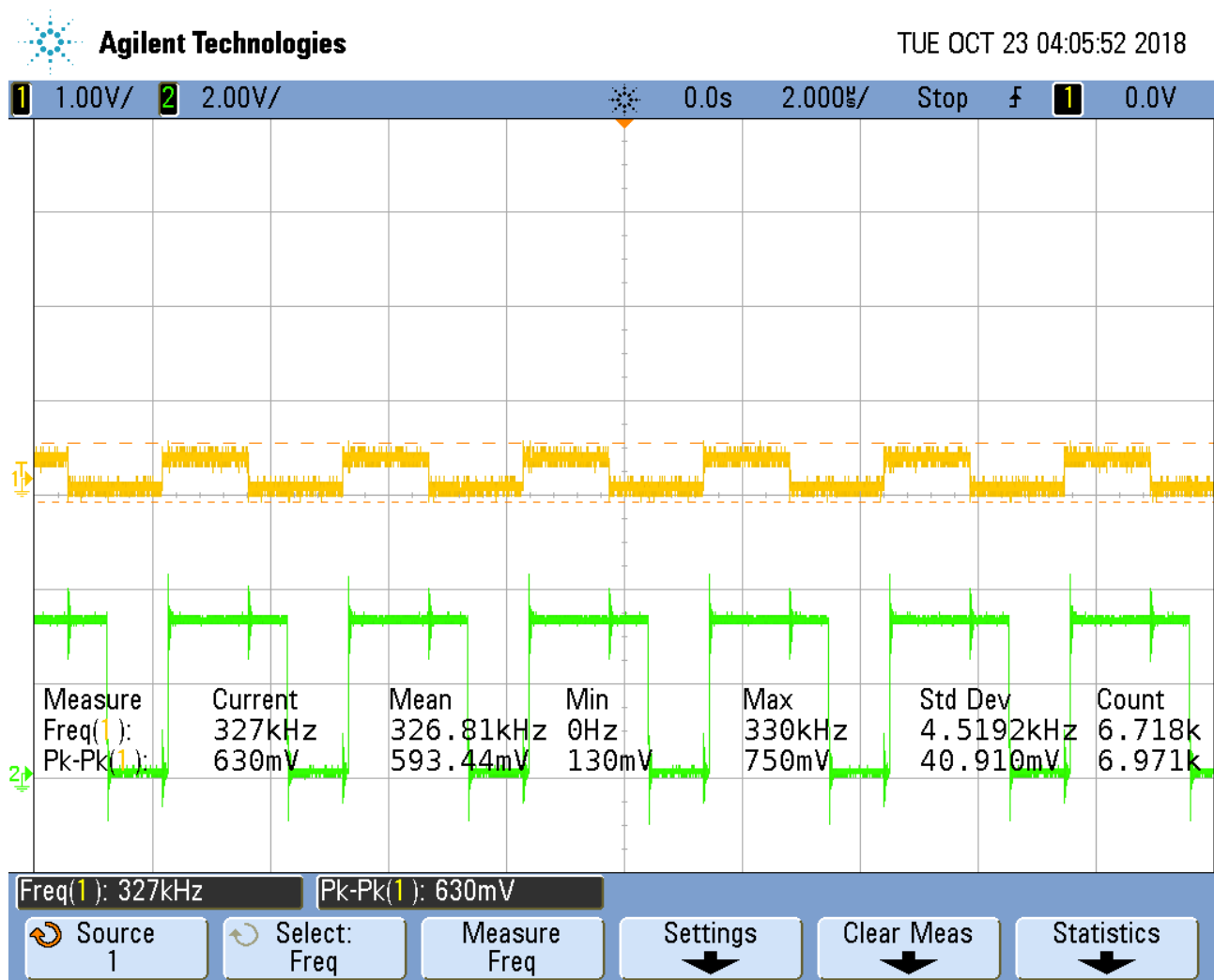
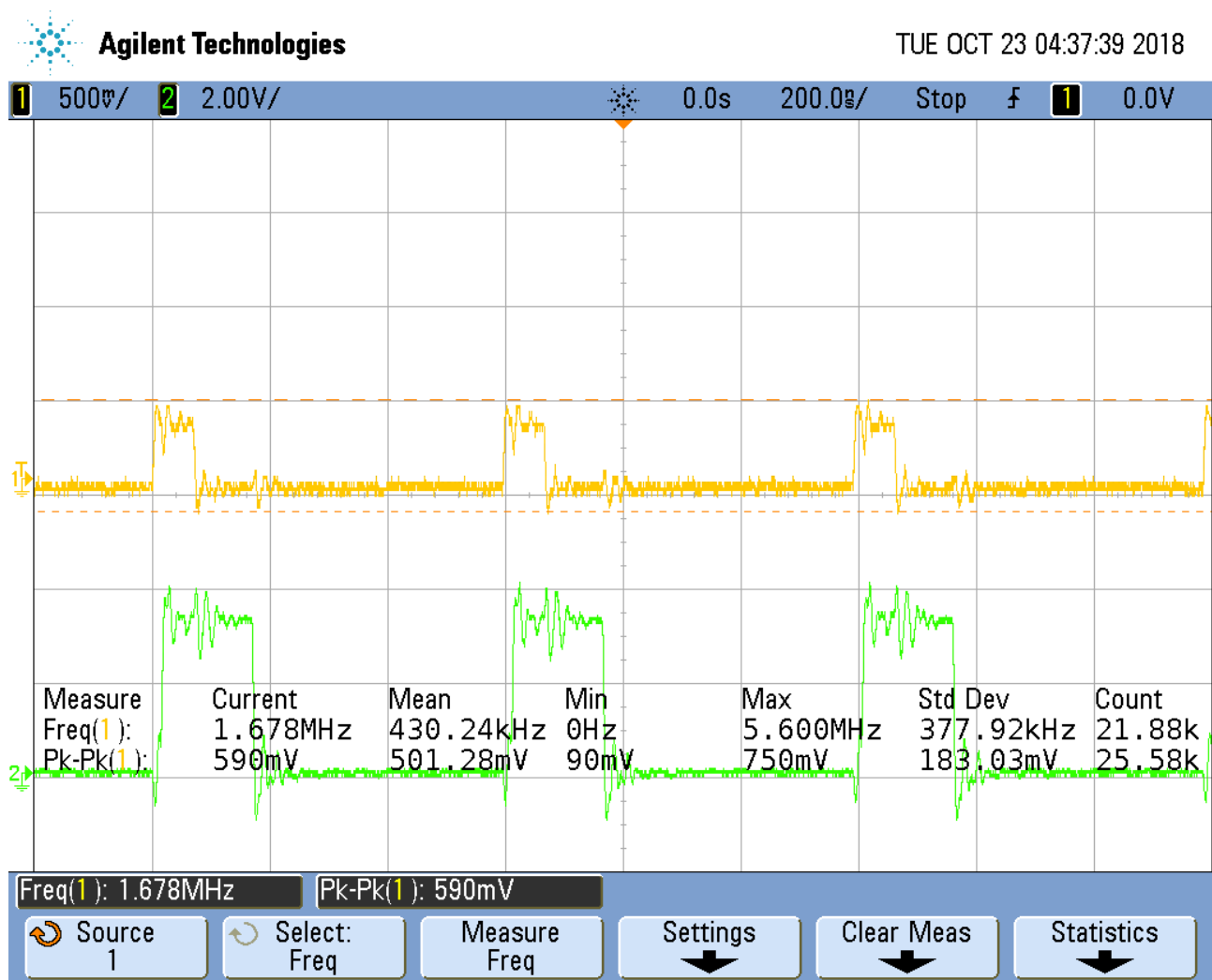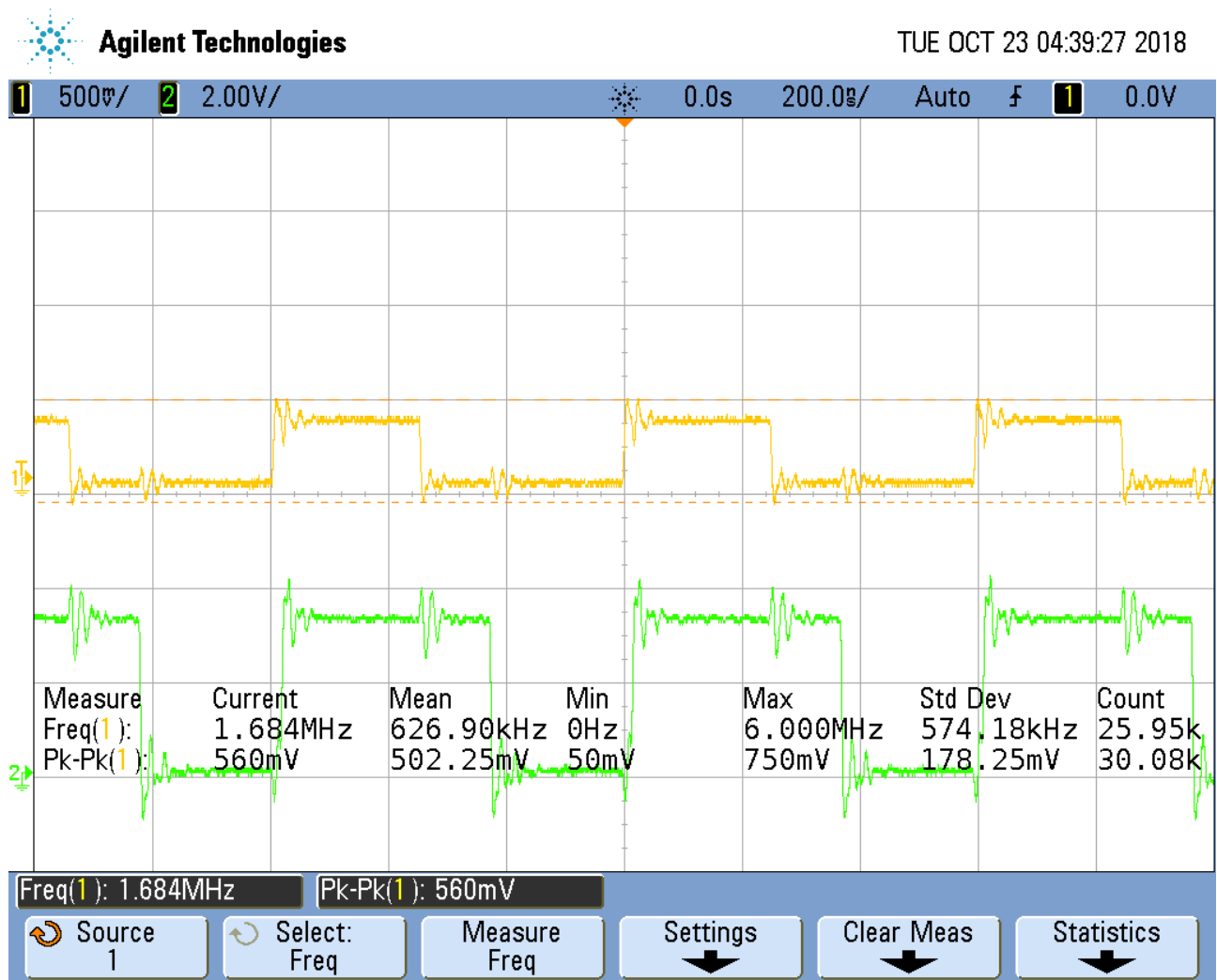Highest frequency with 4 channels: 32.7 MHz, but for showing used other resultion up to 5,6MHz
Jitter: Can see Jitter waves- on both all waveform.
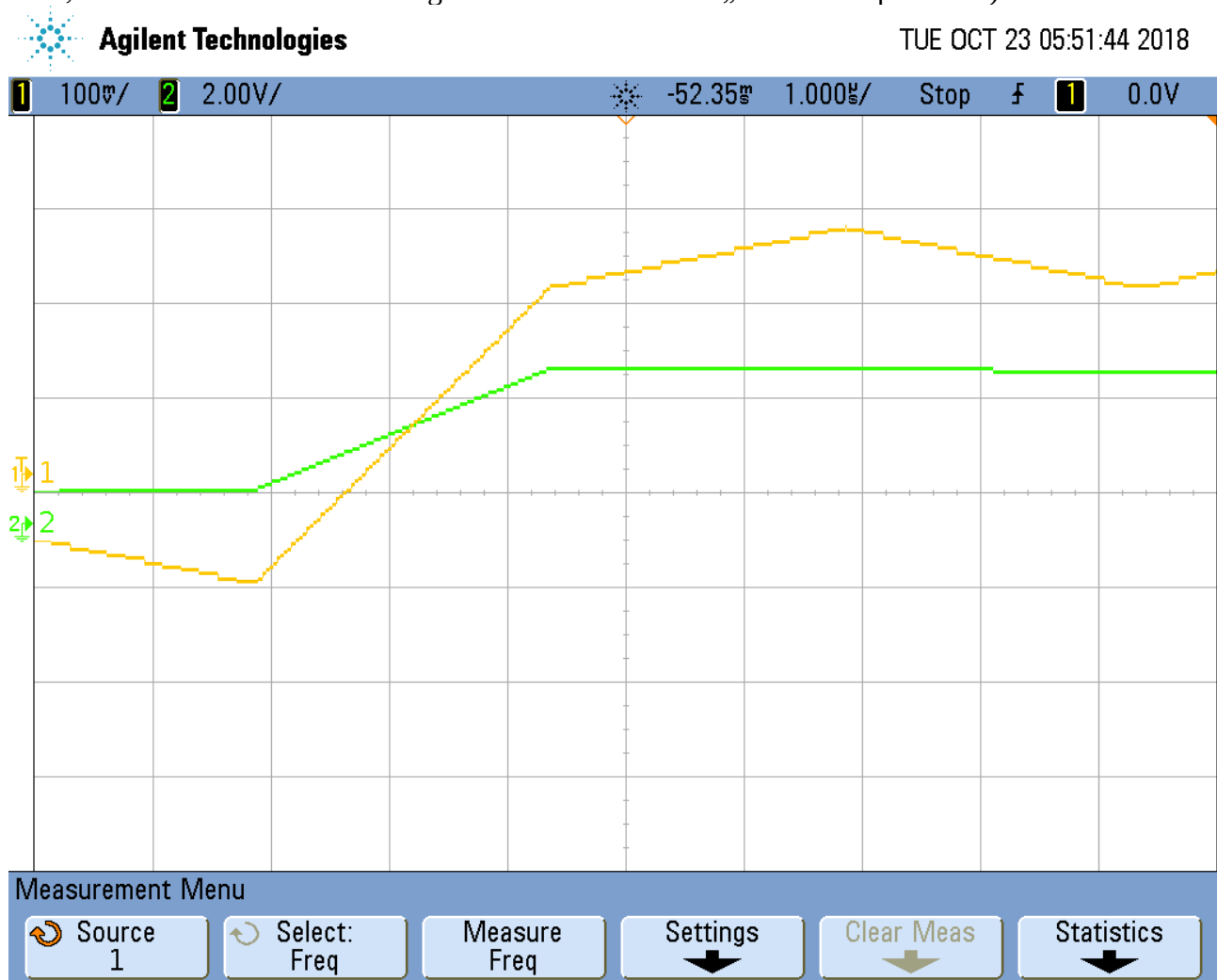
5.5 Loop Unrolling for Better Performance:

Looping speeded the process a little bit up ! See frequency.

5.9 Reading an Input at Regular Intervals:

For this part I used pins P9_25 and P9_27, because I started getting pinmux errors from P9_29.
Also, since I don't have a function generator I tried to used „human manpower". :)

Input Latency: 13.7ns
Measurement weren't really accurate i think the table looks different with using a functional
generator with a increasing frequency