

02-2 Analog Out

Controlling the Brightness of an LED

b.analogWrite()

```
var b = require('bonescript');  
var LED = 'P9_21'; // Pin to use  
var brightness = 0.5;  
  
b.pinMode(LED, b.ANALOG_OUTPUT);  
  
b.analogWrite(LED, brightness);
```

PWM pins

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	EHRPWM1A
GPIO_48	15	16	EHRPWM1B
GPIO_5	17	18	GPIO_4
I2C2_SCL	19	20	I2C2_SDA
EHRPWMOB	21	22	EHRPWMOA
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	ECAPPWM2
EHRPWMOB	29	30	GPIO_122
EHRPWMOA	31	32	VDD_ADC
AIN4	33	34	GND_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	ECAPPWMO
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
TIMER4	7	8	TIMER7
TIMER5	9	10	TIMER6
GPIO_45	11	12	GPIO_44
EHRPWM2B	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
EHRPWM2A	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	EHRPWM1B
GPIO_8	35	36	EHRPWM1A
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
EHRPWM2A	45	46	EHRPWM2B

Up to 8 digital I/O pins can be configured with pulse-width modulators (PWM) to produce signals to control motors or create analog voltage levels, without taking up any extra CPU cycles.

Fading an LED

```
var b = require('bonescript');
var LED = 'P9_21'; // Pin to use
var step = 0.02,    // Step size
    min = 0.02,     // dimmest value
    max = 1,        // brightest value
    brightness = min; // Current brightness;

b.pinMode(LED, b.ANALOG_OUTPUT);

setInterval(fade, 20); // Step every 20 ms

function fade() {
    b.analogWrite(LED, brightness);
    brightness += step;
    if(brightness >= max || brightness <= min) {
        step = -1 * step;
    }
}
```

02-2 Pulse Width Modulation

Controlling an output pin without using the CPU



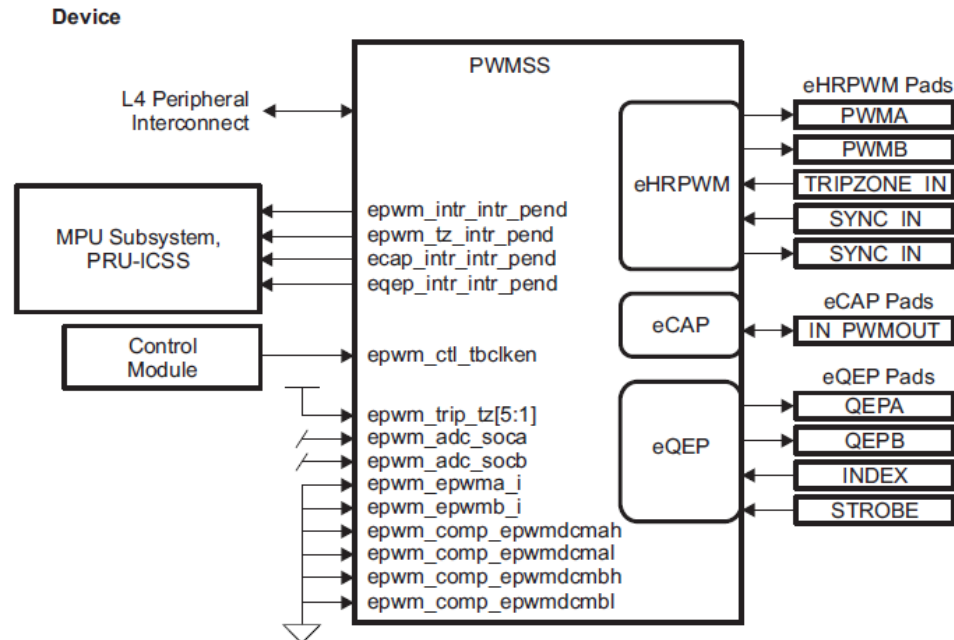
Pulse Width Modulation

- Using the CPU to toggle an IO pin is a poor use of the CPU
- A 1 GHz processor can only toggle at about
 - 100 Hz using the shell, or
 - _____ using a C program
- Many applications could use such a signal
 - at a higher frequency
 - without using so much of the CPU
- Use PWM hardware

PWM Hardware

- The AM335x has a Pulse Width Modulation SubSystem (PWMSS)
- Discussed in Section 17 of the TRM.
- 2 to 4 PWM signals can be produced.

Figure 15-1. PWMSS Integration



Pulse Width Modulation

- The Bone has many standard interfaces
 - i2c, SPI, UART, etc.
- Let's play with the PWM
- Must have same period

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUTTON	9	10	SYS_RESETN
GPIO_30	11	12	GPIO_30
GPIO_31	13	14	EHRPWM1A
GPIO_48	15	16	EHRPWM1B
GPIO_4	17	18	GPIO_5
I2C2_SCL	19	20	I2C2_SDA
EHRPWM0B	21	22	EHRPWM0A
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_125	27	28	ECAPPWM2
EHRPWM0B	29	30	GPIO_122
EHRPWM0A	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	ECAPPWM0
DGND	43	44	DGND
DGND	45	46	DGND

Pin MUXing

- **Problem:** AM335x has more internal lines than hardware IO pins.
- **Solution:** IO pins run through a MUX which selects which internal lines appear on IO pins
- A pin can have 1 from as many as 8 lines assigned to it
- Handled through Device Tree Overlays
- Bonescript handles the Device Tree Overlays

PWM – From Shell

```
bone$ cd /sys/devices/ocp.3
```

```
root@yoder-debian-bone: /sys/devices/ocp.3
root@yoder-debian-bone:/sys/devices/ocp.3# cd /sys/devices/ocp.3/
root@yoder-debian-bone:/sys/devices/ocp.3# ls
44e07000.gpio      48048000.timer    48310000.rng      modalias
44e09000.serial    4804a000.timer    49000000.edma     nop-phy.6
44e0b000.i2c       4804c000.gpio     4a100000.ethernet nop-phy.7
44e10448.bandgap   4819c000.i2c      53100000.sham     nxptda@0.12
44e35000.wdt       481ac000.gpio     53500000.aes      power
44e3e000.rtc       481ae000.gpio     56000000.sgx      rstctl.4
47400000.usb       48200000.interrupt-controller bs_pwm_test_P9_14.15 sound.14
48038000.mcasp     48300000.epwmss   gpio-leds.8       subsystem
48042000.timer     48302000.epwmss   hdmi.13           uevent
48044000.timer     48304000.epwmss   mmc.11
48046000.timer     4830e000.fb       mmc.5
```

PWM (units ns)

```
bone$ cd bs_pwm_test_P9_14.15
```

```
root@yoder-debian-bone: /sys/devices/ocp.3/bs_pwm_test_P9_14.15
780000
root@yoder-debian-bone:/sys/devices/ocp.3/bs_pwm_test_P9_14.15# echo 2 > period
-bash: echo: write error: Invalid argument
root@yoder-debian-bone:/sys/devices/ocp.3/bs_pwm_test_P9_14.15# cd ..
root@yoder-debian-bone:/sys/devices/ocp.3# cd bs_pwm_test_P9_14.15/
root@yoder-debian-bone:/sys/devices/ocp.3/bs_pwm_test_P9_14.15# ls
driver  duty  modalias  period  polarity  power  run  subsystem  uevent
root@yoder-debian-bone:/sys/devices/ocp.3/bs_pwm_test_P9_14.15# cat period
1000000
root@yoder-debian-bone:/sys/devices/ocp.3/bs_pwm_test_P9_14.15# cat duty
780000
root@yoder-debian-bone:/sys/devices/ocp.3/bs_pwm_test_P9_14.15# echo 1000000000 > period
root@yoder-debian-bone:/sys/devices/ocp.3/bs_pwm_test_P9_14.15# echo 500000000 > duty
root@yoder-debian-bone:/sys/devices/ocp.3/bs_pwm_test_P9_14.15#
```

```
bone$ echo 1000000000 > period
```

```
bone$ echo 500000000 > duty
```

PWM

- Units are *ns*
- Try a 1Hz frequency with a 25% duty cycle

```
beagle$ echo 1000000000 > period
```

```
beagle $ echo 250000000 > duty
```

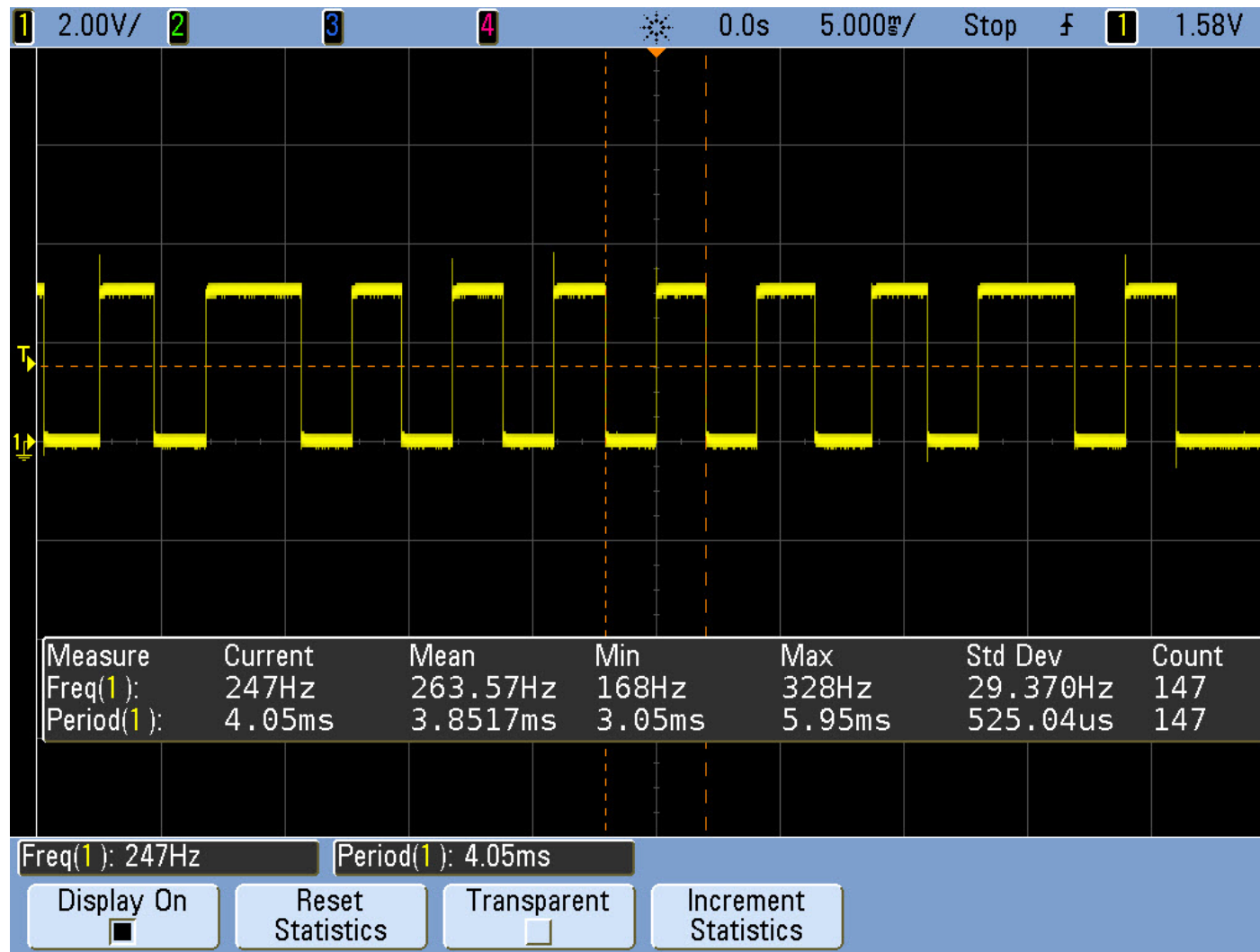
```
beagle $ echo 1 > run
```

- It should be blinking!

Performance

- How fast can the Bone handle I/O?
- I wrote a program to toggle a bit
 - BoneScript
 - Shell
 - C

Performance - BoneScript

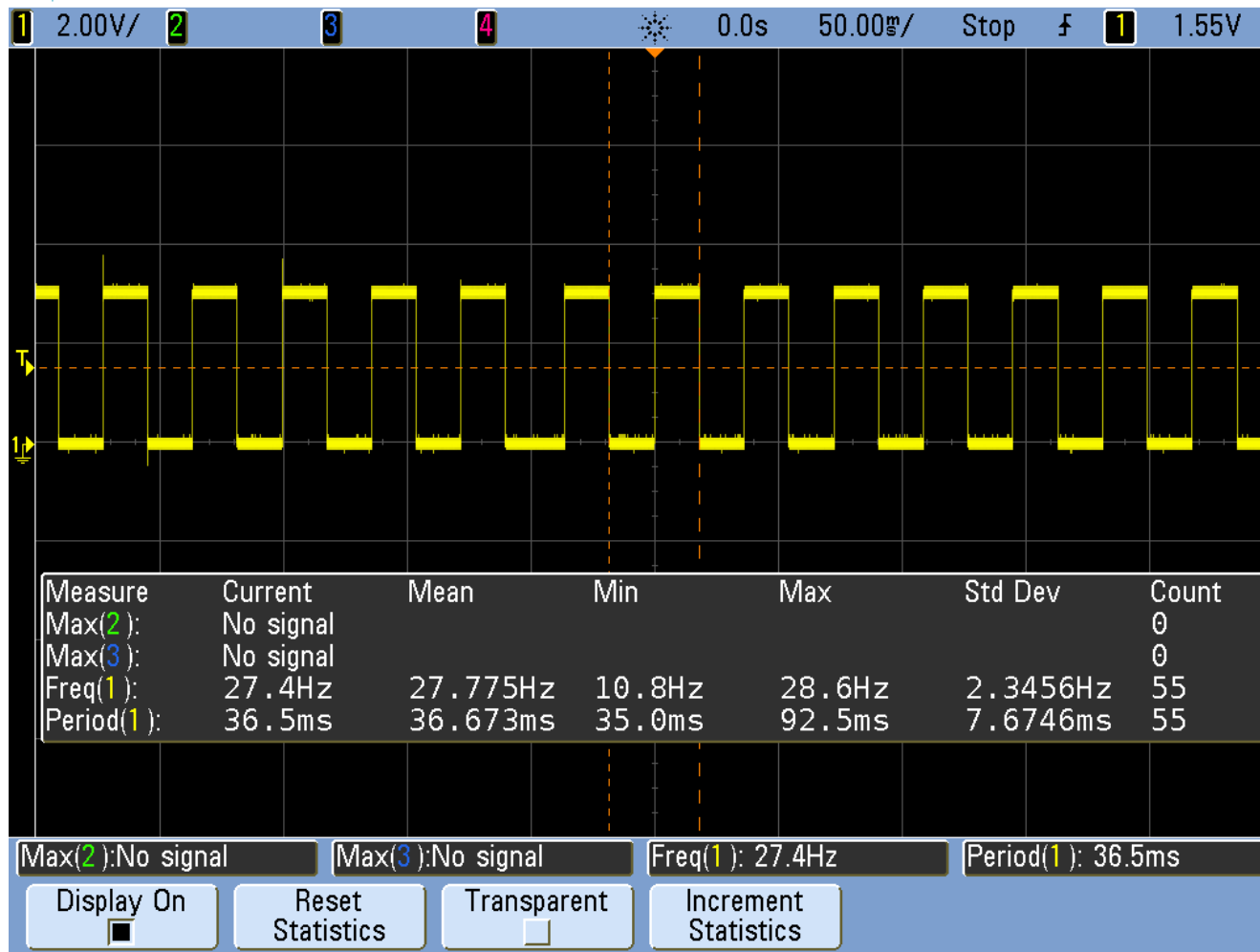


Performance - Shell

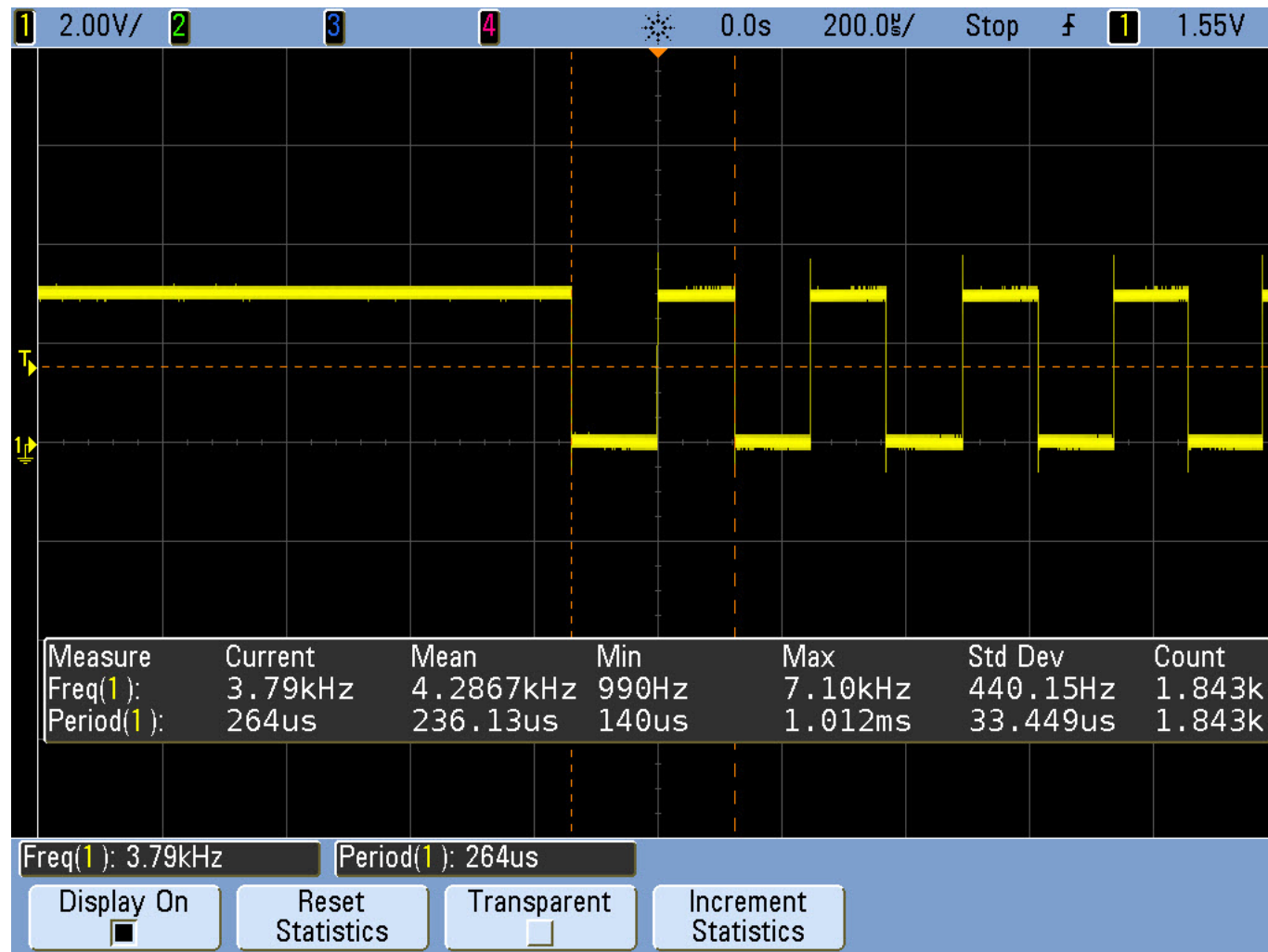


Agilent Technologies

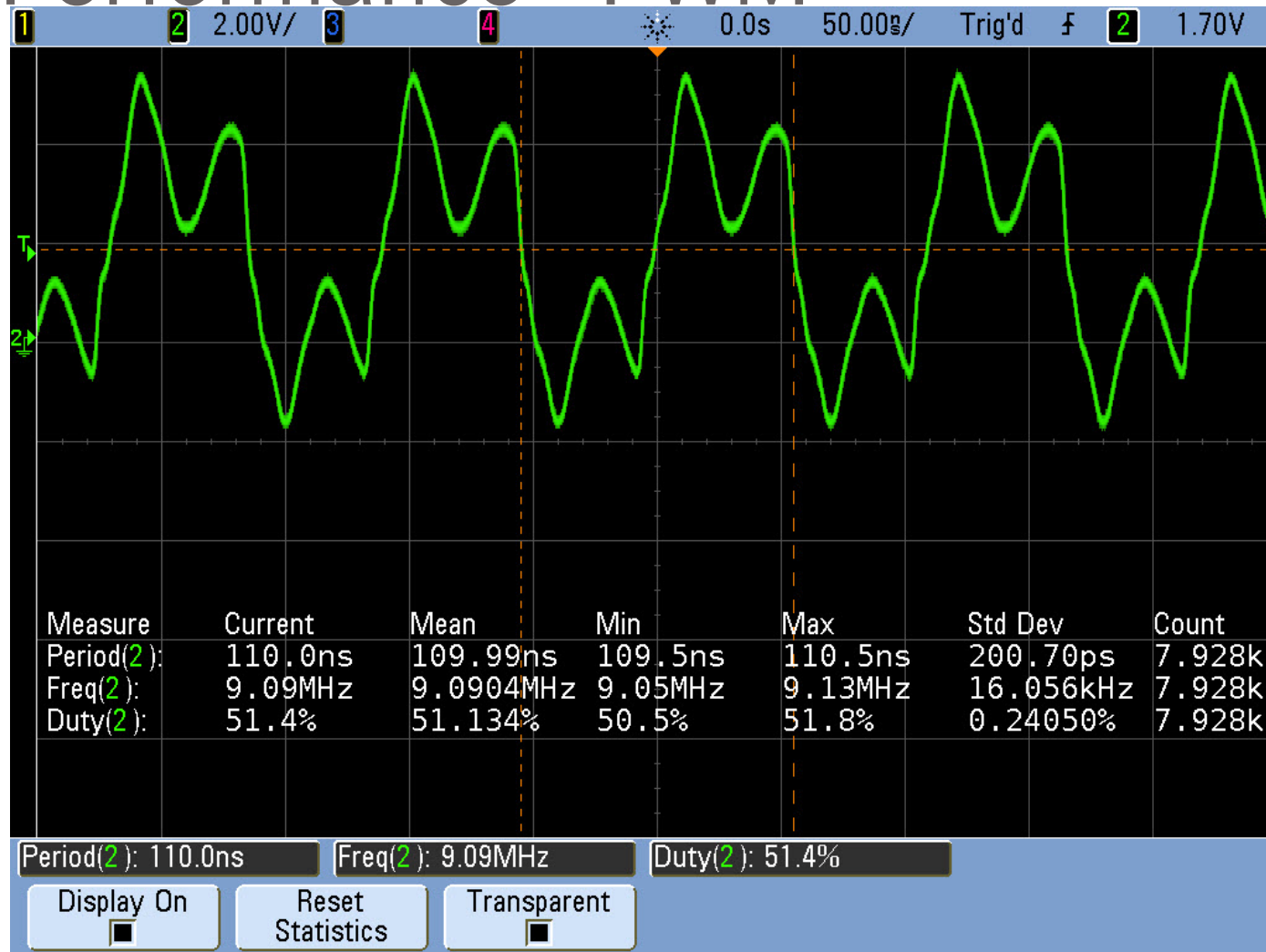
FRI MAY 31 20:18:25 2013



Performance - C



Performance - PWM



Performance - Summary

Language	CPU (%)	Mean (ms)	Min (ms)	Max (ms)
BoneScript	40	3.9	3.0	6.0
Shell	52	37	92	93
C	17	0.24	0.14	1.0
PWM	0	109.99 (ns)	109.5(ns)	110.5(ns)