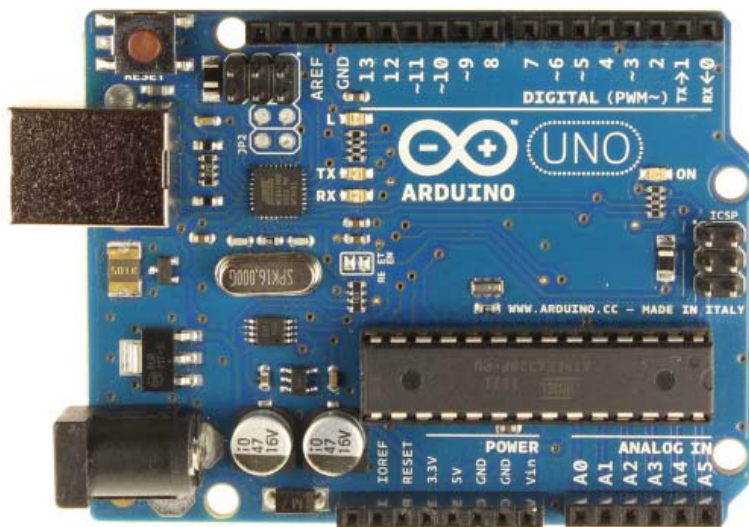


PWM





Pulse Width Modulation

0% Duty Cycle - `analogWrite(0)`



25% Duty Cycle - `analogWrite(64)`



50% Duty Cycle - `analogWrite(127)`



75% Duty Cycle - `analogWrite(191)`



100% Duty Cycle - `analogWrite(255)`



TENSÃO MÉDIA

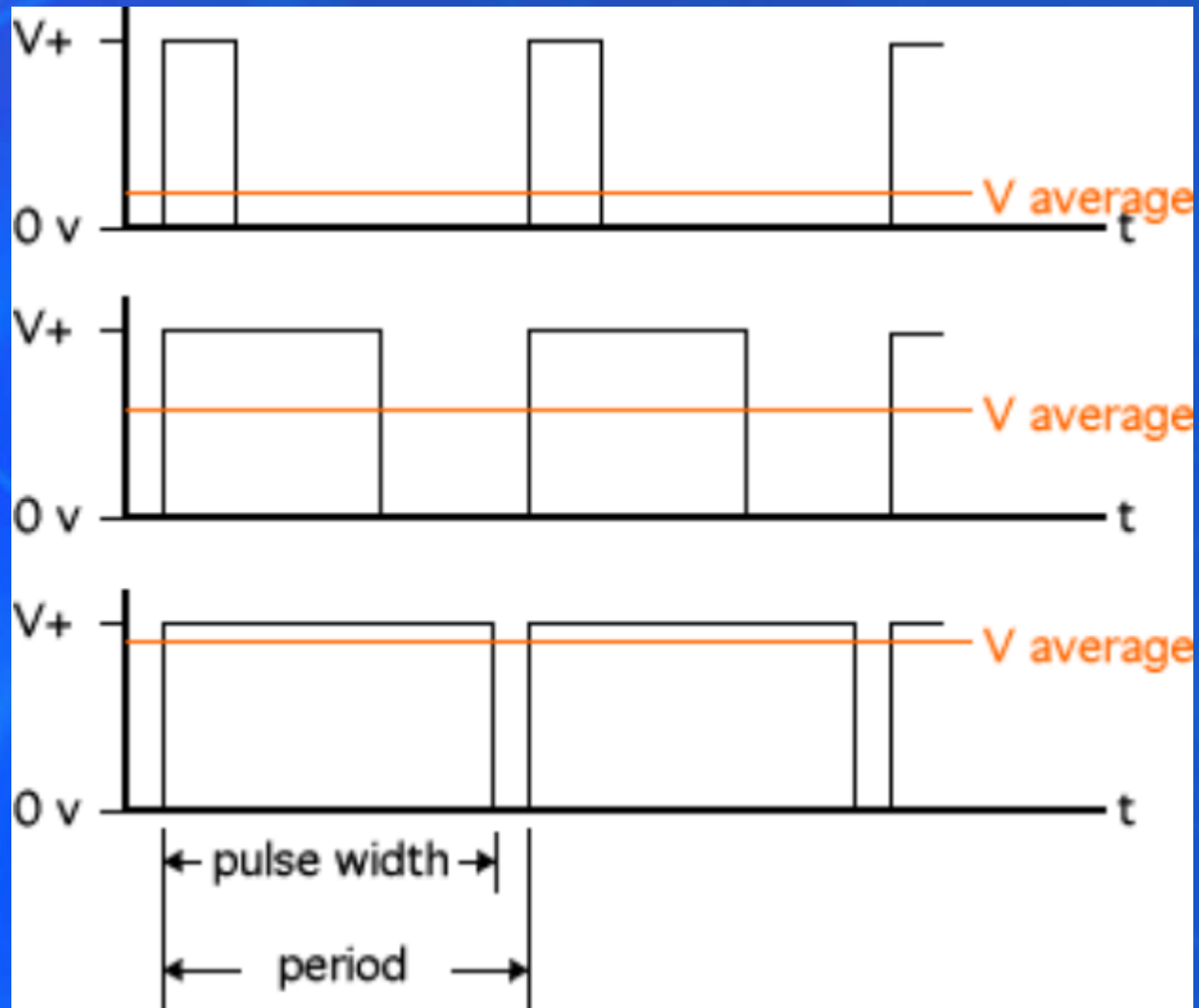
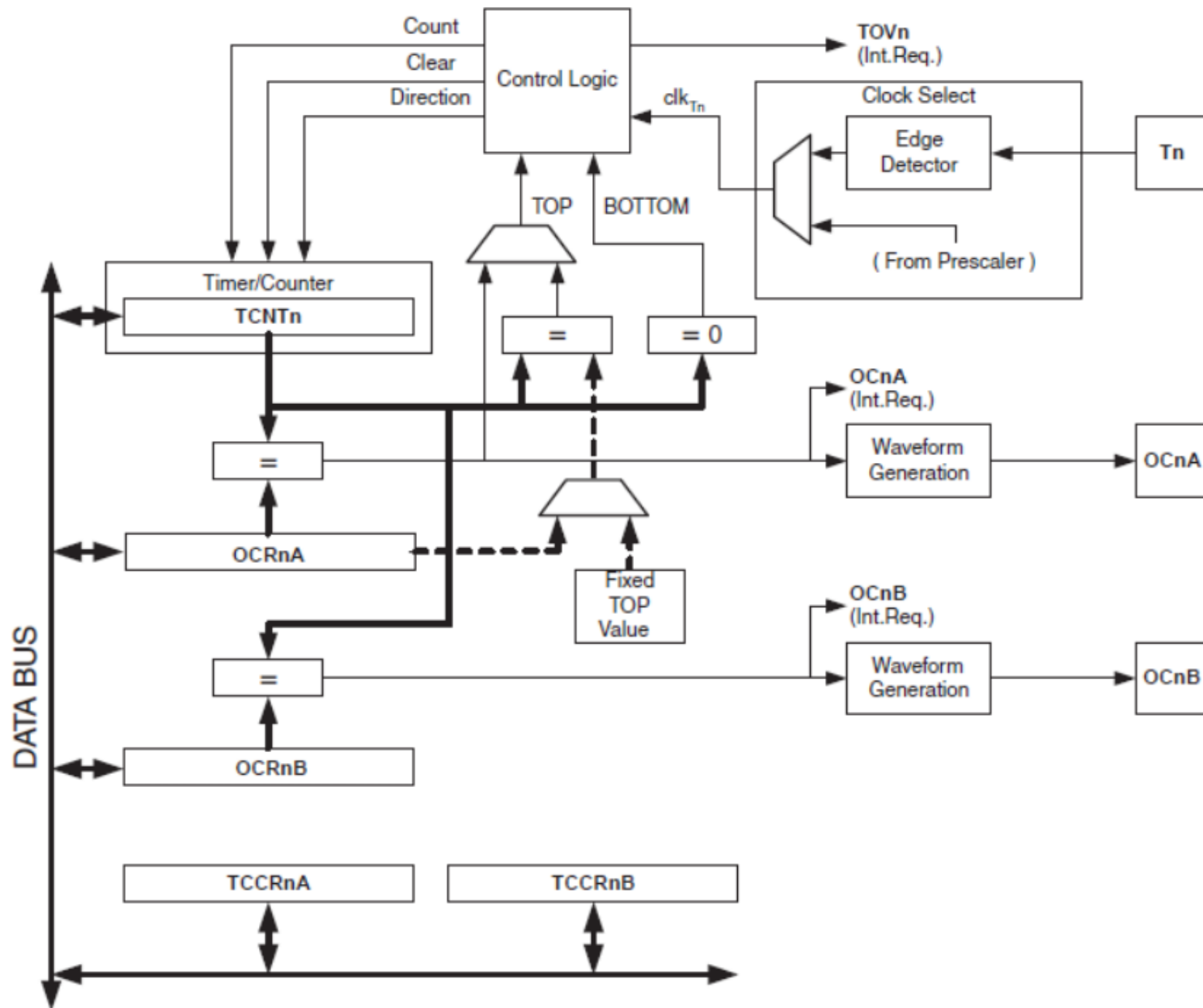
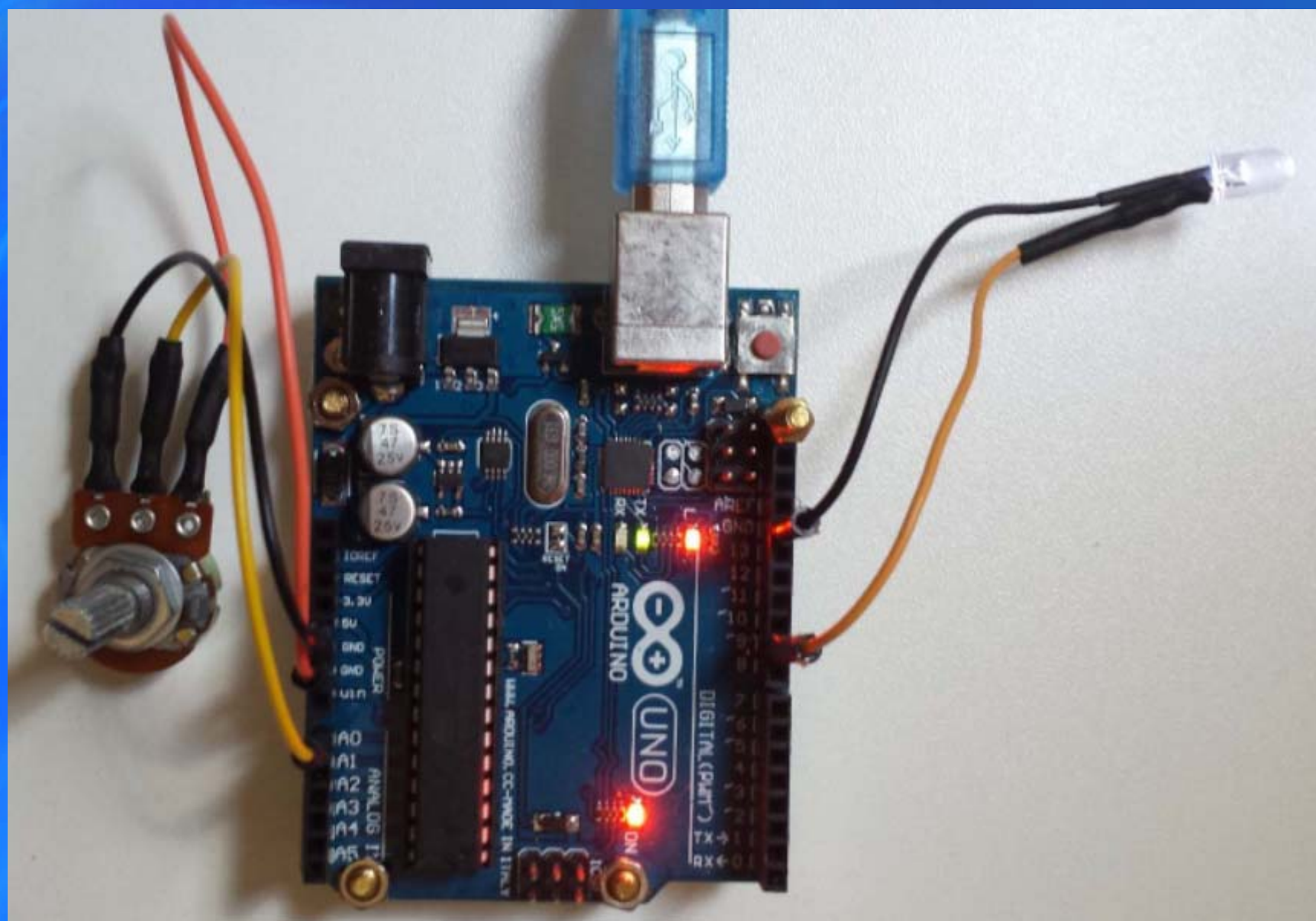


Figure 12-1. 8-bit Timer/Counter Block Diagram





```

10 int ad = 0;
11 int temp = 0;
12 char snum[10];
13
14
15 void pwm_init(void) {
16
17     DDRB |= (1 << DDB1);    // PB1 is now output - OC1A
18     TCCR1A |= (1 << COM1A1); // overrides the normal port functionality of the I/O pin to OC1A - Output Compare Pin
19                             // Pino PB1 do uC e Pino 9 do Kit Arduino (Pag. 131).
20
21     TCCR1A |= (1 << WGM11) | (1 << WGM10); // Timer/Counter Mode of Operation = Fast PWM-10-bit, TOP = 0x03FF,
22                                           // Update of OCR1x at = BOTTOM, TOV1 Flag Set on = TOP (Pag. 132)
23
24     TCCR1B |= (1 << WGM12); // Timer/Counter Mode of Operation = Fast PWM-10-bit, TOP = 0x03FF,
25                             // Update of OCR1x at = BOTTOM, TOV1 Flag Set on = TOP (Pag. 132)
26
27     TCCR1B |= (1 << CS11); // set prescaler to 8 and starts PWM
28 }

```

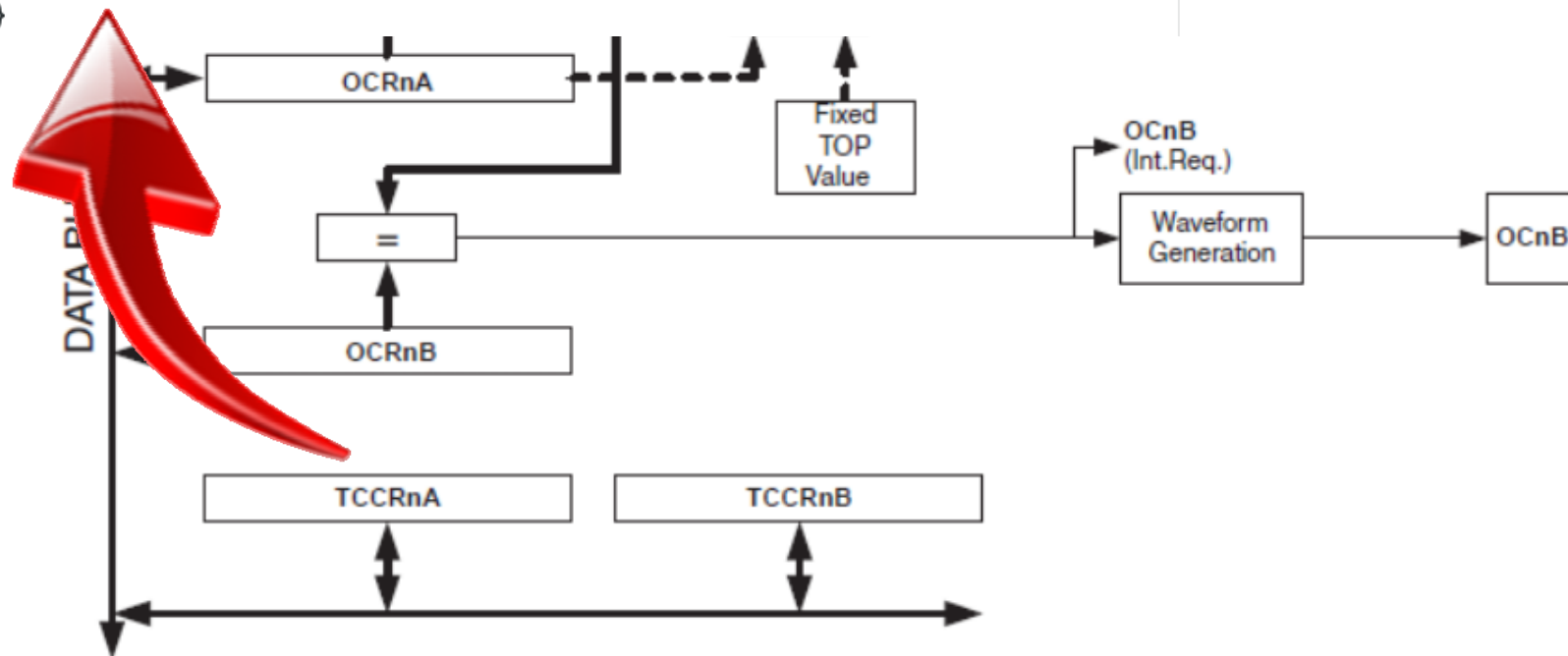
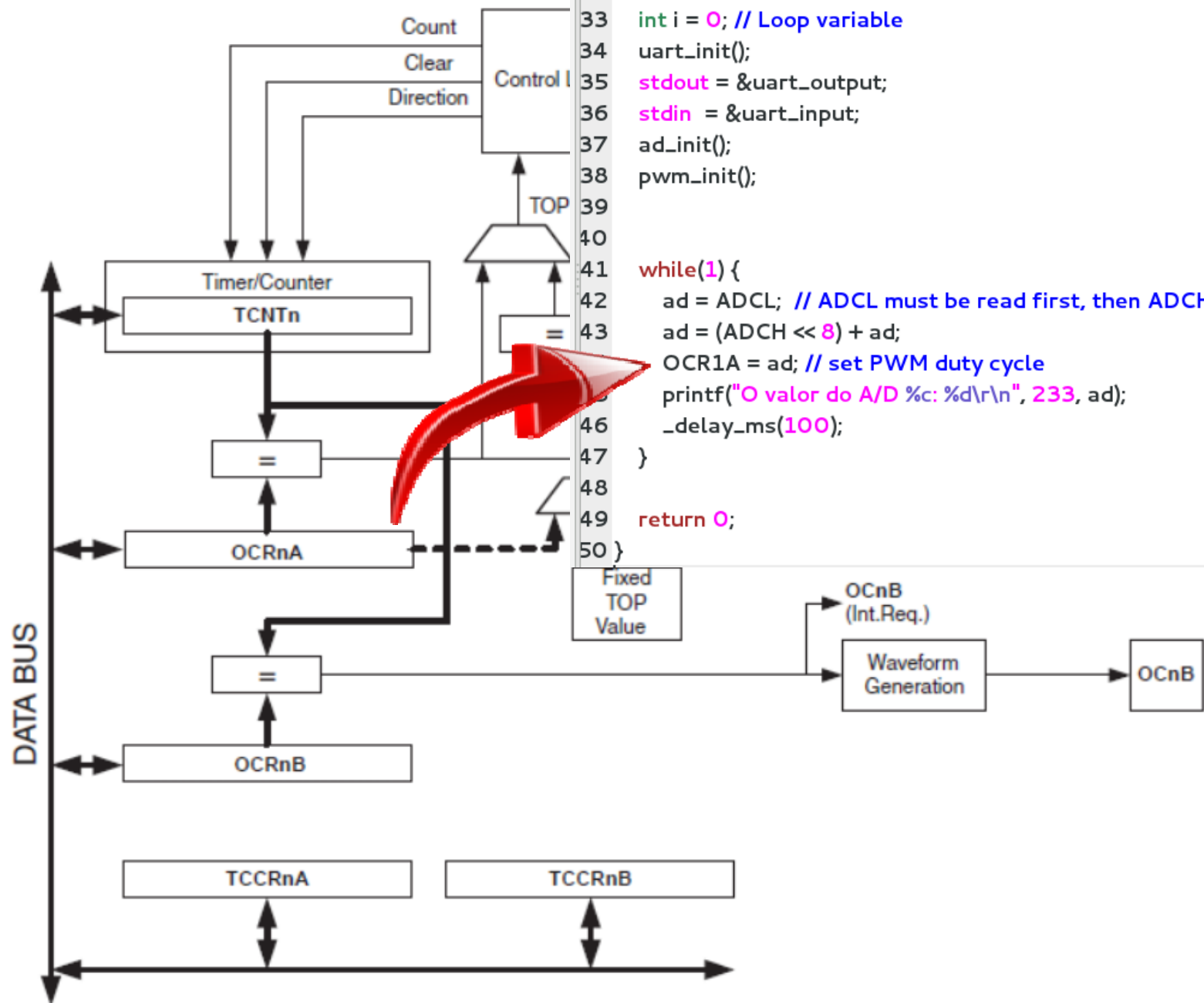


Figure 12-1. 8-bit Timer/Counter Block Diagram





BeagleBone Black

Cape Expansion Headers

P9				P8			
DGND	1	2	DGND	DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3	MMC1_DAT6	3	4	MMC1_DAT7
VDD_5V	5	6	VDD_5V	MMC1_DAT2	5	6	MMC1_DAT3
SYS_5V	7	8	SYS_5V	GPIO_66	7	8	GPIO_67
PWR_BTN	9	10	SYS_RESETN	GPIO_69	9	10	GPIO_68
UART4_RXD	11	12	GPIO_60	GPIO_45	11	12	GPIO_44
UART4_TXD	13	14	EHRPWM1A	EHRPWM2B	13	14	GPIO_26
GPIO_48	15	16	EHRPWM1B	GPIO_47	15	16	GPIO_46
SPI0_CS0	17	18	SPI0_D1	GPIO_27	17	18	GPIO_65
I2C2_SCL	19	20	I2C2_SDA	EHRPWM2A	19	20	MMC1_CMD
SPI0_D0	21	22	SPI0_SCLK	MMC1_CLK	21	22	MMC1_DAT5
GPIO_49	23	24	UART1_TXD	MMC1_DAT4	23	24	MMC1_DAT1
GPIO_117	25	26	UART1_RXD	MMC1_DAT0	25	26	GPIO_61
GPIO_115	27	28	SPI1_CS0	LCD_VSYNC	27	28	LCD_PCLK
SPI1_D0	29	30	GPIO_112	LCD_HSYNC	29	30	LCD_AC_BIAS
SPI1_SCLK	31	32	VDD_ADC	LCD_DATA14	31	32	LCD_DATA15
AIN4	33	34	GNDA_ADC	LCD_DATA13	33	34	LCD_DATA11
AIN6	35	36	AIN5	LCD_DATA12	35	36	LCD_DATA10
AIN2	37	38	AIN3	LCD_DATA8	37	38	LCD_DATA9
AIN0	39	40	AIN1	LCD_DATA6	39	40	LCD_DATA7
GPIO_20	41	42	ECAPPWM0	LCD_DATA4	41	42	LCD_DATA5
DGND	43	44	DGND	LCD_DATA2	43	44	LCD_DATA3
DGND	45	46	DGND	LCD_DATA0	45	46	LCD_DATA1

LEGEND

POWER/GROUND/RESET

AVAILABLE DIGITAL

AVAILABLE PWM

SHARED I2C BUS

RECONFIGURABLE DIGITAL

ANALOG INPUTS (1.8V)

15.1.2 Integration

The Pulse Width Modulation Subsystem (PWMSS) includes a single instance of the Enhanced High Resolution Pulse Width Modulator (eHRPWM), Enhanced Capture (eCAP), and Enhanced Quadrature Encoded Pulse (eQEP) modules. This device includes three instantiations of the PWMSS.

Figure 15-1. PWMSS Integration

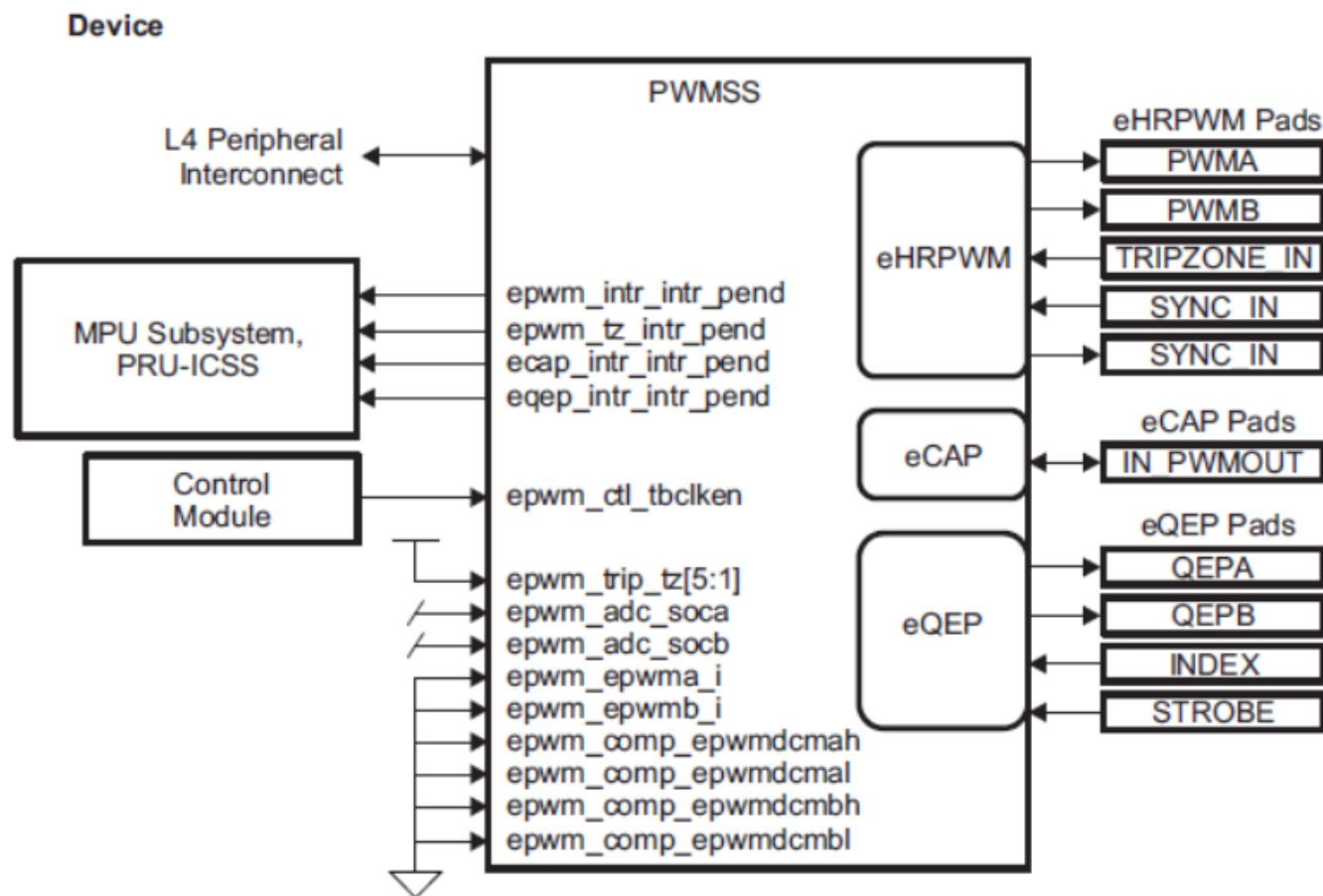
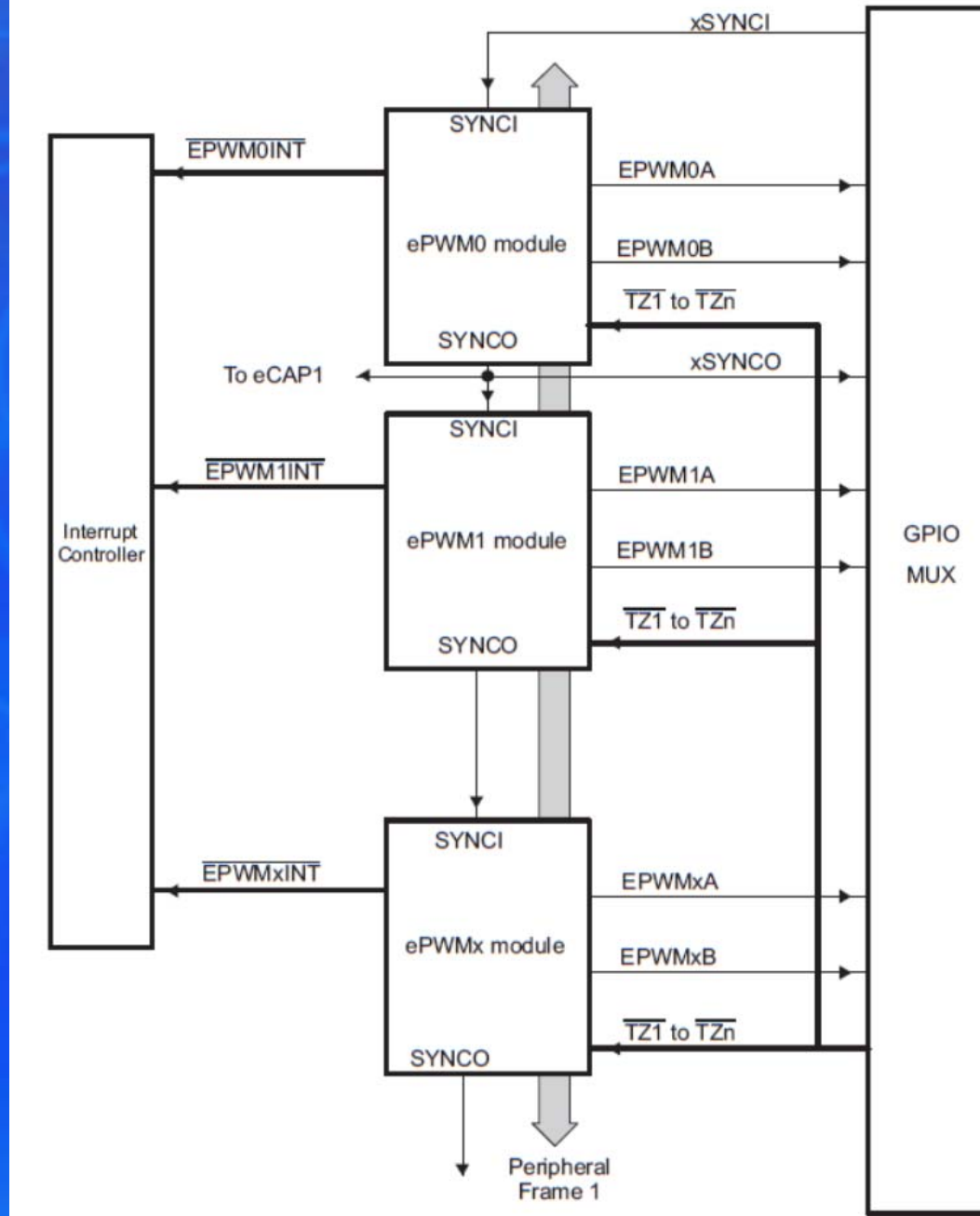


Figure 15-7. Multiple ePWM Modules



EHRPWM0A	1	P9_31	P9_22
EHRPWM0B	2	P9_21	P9_29
EHRPWM1A	3	P9_14	P8_36
EHRPWM1B	4	P9_16	P8_34
EHRPWM2A	5	P8_19	P8_45
EHRPWM2B	6	P8_13	P8_46
ECAPPWM0	7	P9_42	
ECAPPWM2	8	P9_28	

8 PWMs and 4 timers

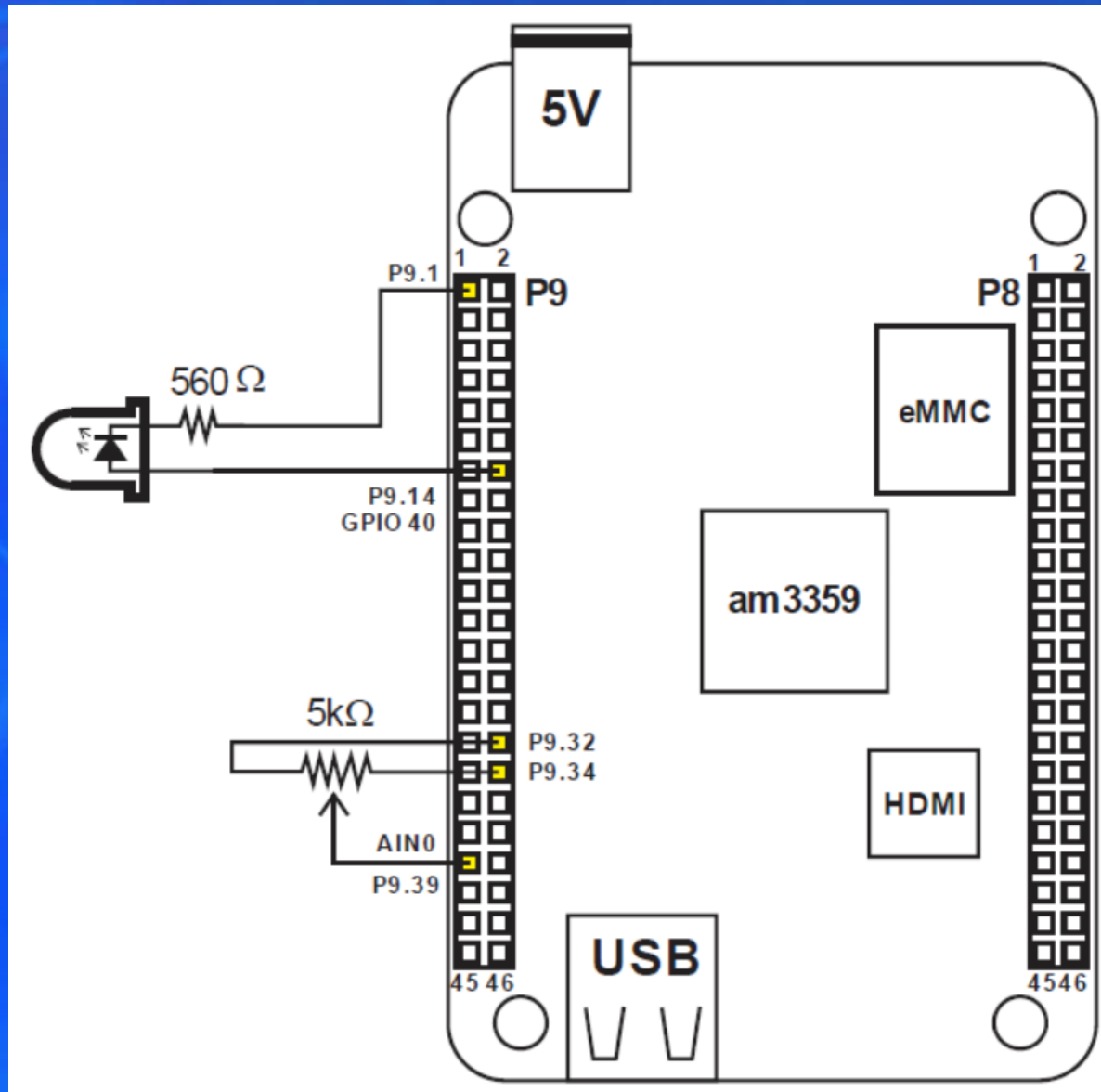
P9

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
EHRPWM0B	21	22	EHRPWM0A
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	ECAPPWM2
EHRPWM0B	29	30	GPIO_112
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

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

Exemplo PWM



```
8 #define SYSFS_ADC_DIR "/sys/bus/iio/devices/iio:device0"
9 #define SYSFS_PWM_DIR "/sys/devices/ocp.3/pwm_test_"
10 #define MAX_BUF 64
11 #define P9_14 "P9_14.16"
12 // O valor 16 do 14.16 deve ser configurado de acordo com o valor do arquivo
13 // (pwm_test_P9_14.x) encontrado na pasta "/sys/devices/ocp.3/"
14 // O valor de ocp.x também pode variar, por isso verifique o valor de x na pasta "/sys/devices/"
15
16 #define PERIOD 409500 //Frequency = (1x10E9) = 1Hz
17                     //Frequency = 1x10E9/409500 = 2442Hz
18 #define DUTY 0      // Duty Depends on Polarity
19 #define POLARITY 0  // Polarity 0 = duty = 0 = 0% duty cycle and duty = PERIOD = 100% duty cycle
20                     // Polarity 1 = duty = PERIOD = 0% duty cycle and duty = 0 = 100% duty cycle
21
22
23 int adc_read(int);
24 void configPWM(const char*, unsigned long, unsigned long, int);
25 void setDuty(const char*, unsigned long);
26
27 int main()
28 {
29     int valorAD;
```

```
103 // Função para atualizar o Duty Cycle
104 void setDuty(const char* pino, unsigned long duty)
105 {
106     int fd, len;
107     char buf[MAX_BUF];
108     snprintf(buf, sizeof(buf), SYSFS_PWM_DIR "%s/duty", pino);
109     fd = open(buf, O_WRONLY);
110     len = snprintf(buf, sizeof(buf), "%lu", duty);
111     write(fd, buf, len);
112     close(fd);
113 }
```



A terminal window titled 'ufu@SE: ~' with a menu bar containing 'Arquivo', 'Editar', 'Ver', 'Pesquisar', 'Terminal', and 'Ajuda'. The terminal shows the following commands and output:

```
root@beaglebone:~# cd /sys/devices/ocp.3/pwm_test_P9_14.16/
root@beaglebone:/sys/devices/ocp.3/pwm_test_P9_14.16# ls
driver  modalias  polarity  run          uevent
duty    period    power     subsystem
root@beaglebone:/sys/devices/ocp.3/pwm_test_P9_14.16#
```



```

43 // Função para a leitura do ADC – canal 0 a 6
44 int adc_read(int pin)
45 {
46     int fd;
47     char buf[MAX_BUF];
48     char ch[5];
49
50     snprintf(buf, sizeof(buf), SYSFS_ADC_DIR "/in_voltage%d_raw", pin);
51     fd = open(buf, O_RDONLY);
52
53     if (fd < 0) {
54         perror("adc/get-value");
55     }
56
57     read(fd, ch, 4);
58
59     close(fd);
60     return atoi(ch);
61 }

```

```

ufu@SE: ~
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
root@beaglebone:~# cd /sys/bus/iio/devices/iio:device0
root@beaglebone:/sys/bus/iio/devices/iio:device0# ls
dev          in_voltage2_raw  in_voltage5_raw  name          uevent
in_voltage0_raw  in_voltage3_raw  in_voltage6_raw  power
in_voltage1_raw  in_voltage4_raw  in_voltage7_raw  subsystem
root@beaglebone:/sys/bus/iio/devices/iio:device0#

```

