/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Filename : servo.c

\*Description : Connect a servo to any pin. It will rotate to random angles.

\*Company : SunRobotics Technologies

\*Website : www.sunrobotics.co.in

\*E-mail : support@sunrobotics.co.in(For Any Query)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "pca9685.h"

#include <wiringPi.h>

#include <stdio.h>

#include <stdlib.h>

#define PIN\_BASE 300

#define MAX\_PWM 4096

#define HERTZ 50

/\*\*

\* Calculate the number of ticks the signal should be high for the required amount of time

\*/

int calcTicks(float impulseMs, int hertz)

{

float cycleMs = 1000.0f / hertz;

return (int)(MAX\_PWM \* impulseMs / cycleMs + 0.5f);

}

/\*\*

\* input is [0..1]

\* output is [min..max]

\*/

float map(float input, float min, float max)

{

return (input \* max) + (1 - input) \* min;

}

int main(void)

{

wiringPiSetupGpio();

printf("PCA9685 servo example\n");

printf("Connect a servo to any pin. It will rotate to random angles\n");

// Setup with pinbase 300 and i2c location 0x40

int fd = pca9685Setup(PIN\_BASE, 0x40, HERTZ);

if (fd < 0)

{

printf("Error in setup\n");

return fd;

}

// Reset all output

pca9685PWMReset(fd);

// Set servo to neutral position at 1.5 milliseconds

// (View http://en.wikipedia.org/wiki/Servo\_control#Pulse\_duration)

float millis = 1.5;

int tick = calcTicks(millis, HERTZ);

pwmWrite(PIN\_BASE + 16, tick);

delay(2000);

int active = 1;

while (active)

{

// That's a hack. We need a random number < 1

float r = rand();

while (r > 1)

r /= 10;

millis = map(r, 1, 2);

tick = calcTicks(millis, HERTZ);

pwmWrite(PIN\_BASE + 16, tick);

delay(1000);

}

return 0;

}