**首先要在头文件中定义一个 \_timerview**

**ViewController.h**

@interface ViewController : UIViewController

{

/\*

定义一个定时器对象

可以通过每隔固定时间发送一个消息

可以通过此消息来调用相应的时间函数

通过此函数可以在固定时间段来完成一个根据时间间隔的任务

\*/

NSTimer\* \_timerView;

}

//定时器的属性对象

@property (retain,nonatomic)NSTimer\* timerView;

**ViewController.m**

@implementation ViewController

//属性和成员变量同步

@synthesize timerView = \_timerView;

- (void)viewDidLoad {

[super viewDidLoad];

// Do any additional setup after loading the view, typically from a nib.

UIButton\* btn = [UIButton buttonWithType:UIButtonTypeRoundedRect];

btn.frame = CGRectMake(100, 100, 180, 40);

[btn setTitle:@"按钮定时器启动=w=" forState:UIControlStateNormal];

[btn addTarget:self action:@selector(pressStart) forControlEvents:UIControlEventTouchUpInside];

[self.view addSubview:btn];

//停止定时器按钮

UIButton\* btnStop = [UIButton buttonWithType:UIButtonTypeRoundedRect];

btnStop.frame = CGRectMake(100, 200, 180, 40);

[btnStop setTitle:@"按钮定时器停止=。=" forState:UIControlStateNormal];

[btnStop addTarget:self action:@selector(pressStop) forControlEvents:UIControlEventTouchUpInside];

[self.view addSubview:btnStop];

UIView\* view = [[UIView alloc]init];

view.frame = CGRectMake(0, 0, 80, 80);

view.backgroundColor = [UIColor orangeColor];

[self.view addSubview:view];

//设置view的标签值

//通过父亲视图对象以及view的标签值可以获得相应的视图对象

view.tag = 101;

}

//按下开始按钮函数

-(void)pressStart

{

//NSTimer的类方法创建一个定时器并且启动这个定时器

//P1：每隔多长时间调用定时器函数，以秒为单位

//P2：表示实现定时器函数的对象

//P3：定时器函数对象

//P4：可以定时器函数中一个参数，无参数可以传nil

//P5：定时器是否重复操作 yes为重复 no只完成一次函数调用

//返回值为一个新建好的定时器对象

\_timerView = [NSTimer scheduledTimerWithTimeInterval:0.02 target:self selector:@selector(updateTimer:) userInfo:@"hades" repeats:YES];

}

//定时器函数

//可以将定时器本身作为参数传入

-(void)updateTimer:(NSTimer\*)timer

{

NSLog(@"test name = %@",timer.userInfo);

//最好tag从100开始

UIView\* view = [self.view viewWithTag:101];

view.frame = CGRectMake(view.frame.origin.x+0.5, view.frame.origin.y+0.5, 80, 80);

}

//按下停止按钮时调用

-(void)pressStop

{

//停止定时器

[\_timerView invalidate];

}