```c
/*
 * @brief FreeRTOS Blinky example
 *
 * @note
 * Copyright(C) NXP Semiconductors, 2014
 * All rights reserved.
 *
 * @par
 * Software that is described herein is for illustrative purposes only
 * which provides customers with programming information regarding the
 * LPC products.  This software is supplied "AS IS" without any warranties of
 * any kind, and NXP Semiconductors and its licensor disclaim any and
 * all warranties, express or implied, including all implied warranties of
 * merchantability, fitness for a particular purpose and non-infringement of
 * intellectual property rights.  NXP Semiconductors assumes no responsibility
 * or liability for the use of the software, conveys no license or rights under any
 * patent, copyright, mask work right, or any other intellectual property rights in
 * or to any products. NXP Semiconductors reserves the right to make changes
 * in the software without notification. NXP Semiconductors also makes no
 * representation or warranty that such application will be suitable for the
 * specified use without further testing or modification.
 *
 * @par
 * Permission to use, copy, modify, and distribute this software and its
 * documentation is hereby granted, under NXP Semiconductors' and its
 * licensor's relevant copyrights in the software, without fee, provided that it
 * is used in conjunction with NXP Semiconductors microcontrollers.  This
 * copyright, permission, and disclaimer notice must appear in all copies of
 * this code.
 */

#include "board.h"
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "timers.h"
#include "semphr.h"


void employee_task()
{

}

xSemaphoreHandle employee_signal=0;
```

```c
void boss(void *p)
{
    while(1)
    {
        puts("Boss giving the signal");
        xSemaphoreGive(employee_signal);
        puts("Boss finished giving the signal");
        vTaskDelay(2000);
    }
}
void employee(void *p){
    while(1)
    {
        if (xSemaphoreTake(employee_signal, portMAX_DELAY))
        {
         employee_task();
        puts("employee has finished its task");

    }
}
}
int main(void)
{

    vSemaphoreCreateBinary(employee_signal);
    /* create tasks*/

    xTaskCreate(boss, (signed char*)"t1", 1024,NULL,1,NULL);
    xTaskCreate(employee, (signed char*)"t2", 1024,NULL,2,NULL);

    vTaskStartScheduler();
    return 0;
}


```