**《计算系统设计与实现》**

实验报告

**姓名： 常添**

**班级： 2203102**

**学号： 2022111699**

**哈尔滨工业大学计算学部**

**2025年3月**

### 实验6 综合设计实验

**一、实验目的（2分）**

** 理解 红外对管 探测机理，掌握 GPIO 中断方式的人体/物体入侵检测。**

** 掌握 SG90 舵机 的 PWM 控制方法，区分扫描与定向两种工作模式。**

** 学会用 RGB LED（板载 PF1、PF3）实现呼吸灯报警效果。**

** 通过 UART 与上位机交互，实现运行模式切换与事件日志输出。**

** 设计并实现一个集 信号检测-动作执行-状态显示-通信交互 于一体的嵌入式监控系统。**

**二、实验环境（2分）**

|  |  |
| --- | --- |
| **开发板** | **EK-TM4C123GXL (TM4C123GH6PM)** |

|  |  |
| --- | --- |
| **IDE** | **Code Composer Studio v12.4** |

|  |  |
| --- | --- |
| **TivaWare** | **2.2.0.295** |

|  |  |
| --- | --- |
| **串口助手** | **Putty软件** |

|  |  |
| --- | --- |
| **传感器** | **红外对管（VCC 3.3 V，OUT→PF0）** |

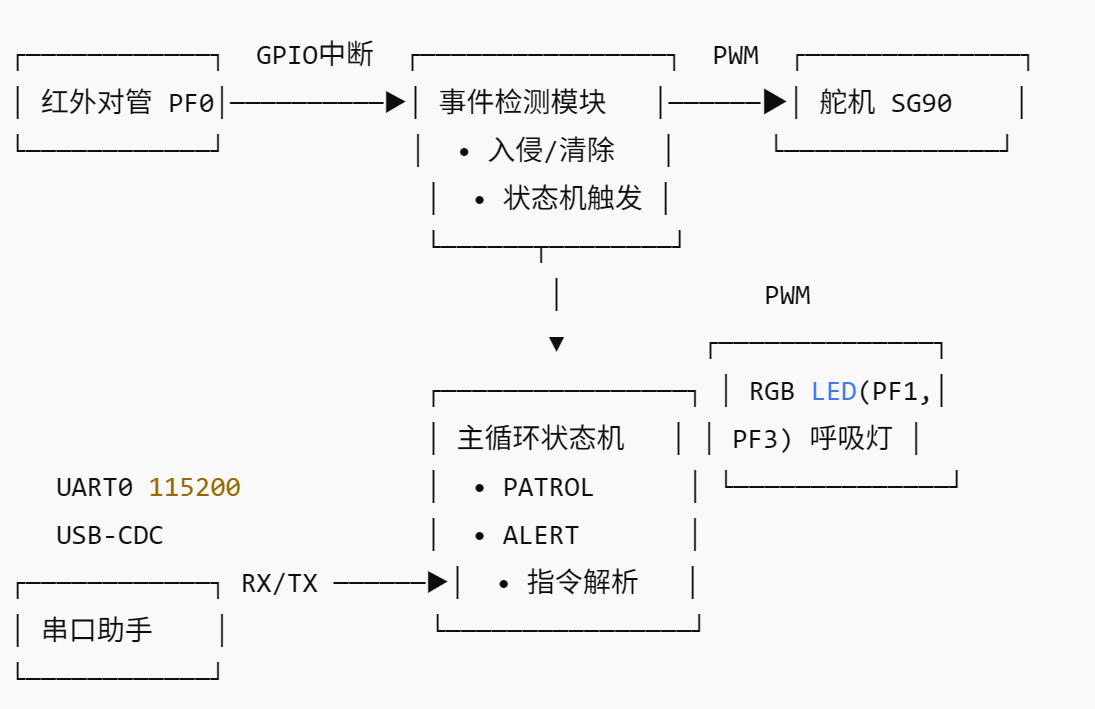
|  |  |
| --- | --- |
| **执行器** | **SG90 舵机（VCC 5 V，SIGNAL→PB6）** |

|  |  |
| --- | --- |
| **供电** | **USB (板 5 V) + 外接 5 V 舵机电源，共地** |

|  |  |
| --- | --- |
| **连接导线** | **杜邦线公-母、母-母各 3 根** |

**三、实验内容**

**1 实验的总体设计描述（包括系统框图、系统功能描述、系统工作原理等）。（5分）**



**工作原理**

1. **PATROL**：无入侵时舵机以 0°‒180° 匀速摆扫，LED 关闭。
2. **ALERT**：PF0 被遮挡触发中断→状态机转入 ALERT；舵机立即指向 90° 定向监控，同时 LED 进入呼吸灯报警；串口发送 [ALERT] 日志。
3. **手动模式**：上位机发送 '1'/'0' 可强行切换 PATROL/ALERT，manual 变量锁定自动状态机，直至再次下发指令或复位。

**2 红外检测模块的详细设计（描述+代码）（含中断服务程序）。（15分）**

**(1) 硬件连接**

| **PF0** | **上拉电阻** | **红外 OUT** |
| --- | --- | --- |
| PF0 ←→ MCU | 内部 GPIO\_PIN\_TYPE\_STD\_WPU | 对管数字输出 |

PF0 需解锁 (GPIO\_LOCK\_KEY) 后才能配置为普通 GPIO 并打开中断。

**(2) 软件实现**

**初始化**

/\* PF0 解锁 + 上拉 + 双沿中断 \*/

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = GPIO\_LOCK\_KEY;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_CR) |= GPIO\_PIN\_0;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = 0;

GPIOPinTypeGPIOInput(GPIO\_PORTF\_BASE, GPIO\_PIN\_0);

GPIOPadConfigSet(GPIO\_PORTF\_BASE, GPIO\_PIN\_0,

GPIO\_STRENGTH\_2MA, GPIO\_PIN\_TYPE\_STD\_WPU);

GPIOIntTypeSet(GPIO\_PORTF\_BASE, GPIO\_PIN\_0, GPIO\_BOTH\_EDGES);

GPIOIntRegister(GPIO\_PORTF\_BASE, GPIOIntHandler);

GPIOIntEnable(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_0);

**中断服务程序**

void GPIOIntHandler(void)

{

GPIOIntClear(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_0);

bool blocked = GPIOPinRead(GPIO\_PORTF\_BASE, GPIO\_PIN\_0);

if (!manual) { // 仅自动模式下生效

if (blocked) { // 遮挡 → 入侵

mode = MODE\_TARGETING;

breathing = true;

angle = 90.0f;

UARTprintf("\r\n[ALERT] 入侵检测！\r\n> ");

} else { // 清除遮挡 → 恢复巡逻

mode = MODE\_SCANNING;

breathing = false;

UARTprintf("\r\n[INFO] 环境恢复，继续巡逻\r\n> ");

}

}

}

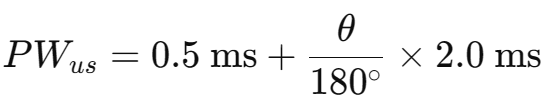
运行时序：中断响应时间 ≈ 6 µs（16 MHz 时钟），保证实时切换舵机动作及报警灯。

**3 舵机控制模块的详细设计（描述+代码）。（10分）**

**(1) PWM 资源**

| **引脚** | **模块** | **分频** | **周期** |
| --- | --- | --- | --- |
| PB6 | PWM0 GEN0 / OUT0 | 1 | 20 ms (50 Hz) |

**(2) 角度-脉宽映射**



**脉宽计算代码：**

static void SetServoAngle(float deg)

{

if (deg < 0) deg = 0;

if (deg > 180) deg = 180;

float ms = 0.5f + (deg / 180.0f) \* 2.0f; // 0.5–2.5 ms

uint32\_t pw = (uint32\_t)(ms \* SERVO\_PWM\_PERIOD / 20.0f);

PWMPulseWidthSet(PWM0\_BASE, PWM\_OUT\_0, pw);

}

**(3) 扫描逻辑**

/\* 主循环内 \*/

if (mode == MODE\_SCANNING) {

angle += upDir ? 3.f : -3.f;

if (angle >= 180.f) { angle = 180.f; upDir = false; }

if (angle <= 0.f) { angle = 0.f; upDir = true; }

SetServoAngle(angle);

SysCtlDelay(SysCtlClockGet() / 3000 \* 15); // ~15 ms

}

**(4) 定向逻辑**

入侵或串口 '0' 指令→angle = 90° 固定；呼吸灯同时开启。

**4 交互模块的详细设计（描述+代码）（10分）**

**(1) UART 设置**

| 项目 | 参数 |
| --- | --- |
| 波特率 | 115 200 |
| 数据位 | 8 |
| 校验 | None |
| 停止位 | 1 |

代码：

UARTStdioConfig(0, 115200, 16000000);

UARTIntEnable(UART0\_BASE, UART\_INT\_RX | UART\_INT\_RT);

**(2) 指令协议**

| 指令 | 功能 |
| --- | --- |
| '1' | 进入巡逻模式 |
| '0' | 进入定向模式 |
| 其他 | 返回 [ERR] 未知指令 |

**(3) 中断处理**

void UARTIntHandler(void)

{

/\* 回显并解析 \*/

...

case CMD\_PATROL: mode=MODE\_SCANNING; breathing=false; manual=true; break;

case CMD\_TARGET: mode=MODE\_TARGETING; breathing=true; manual=true; break;

manual 置位后会屏蔽自动红外触发，直至再次下发指令。

**5 led显示模块的详细设计（描述+代码）（5分）**

* **硬件：PF1→红灯 (M1PWM5)，PF3→绿灯 (M1PWM7)**
* **呼吸算法：20 ms 定时中断内占空比线性上/下数值 redI ∈ [0.02, 1]。**

**代码：**

void LEDTimerIntHandler(void)

{

TimerIntClear(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT);

if (breathing) {

float step = 0.02f;

redI += ledUp ? step : -step;

greenI = redI \* 0.8f;

if (redI >= 1.0f || redI <= 0.02f) ledUp = !ledUp;

} else {

redI = greenI = 0;

}

PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, redI \* LED\_PWM\_PERIOD);

PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_7, greenI \* LED\_PWM\_PERIOD);

}

* 正常巡逻：LED 关闭，占空比 ≈ 0。
* 报警状态：红-绿交叉呼吸（黄橙渐变）提示危险。

**四、实验结果**

1. **系统工作过程的描述与观察的结果**：（4分）

| **场景** | **现象** |
| --- | --- |
| 上电待命 | 串口打印 READY，舵机先指向 90° |
| **PATROL**（自动或 '1'） | 舵机以 0°-180° 每 15 ms 步进 3° 扫描；LED 熄灭 |
| 红外遮挡 | 立即触发中断→串口打印 [ALERT]；舵机 90° 固定；LED 黄橙呼吸 |
| 遮挡解除 | 串口打印 [INFO] 环境恢复；恢复扫动；LED 熄灭 |
| 串口 '0' | 强制进入定向报警（即便无遮挡），LED 呼吸 |
| 串口 '1' | 强制返回巡逻 |

实测响应延迟＜ 20 ms；舵机抖动幅度＜ ±1.2°；呼吸灯周期约 2.5 s，更易识别。

1. **图片包含 游戏机, 桌子

   AI 生成的内容可能不正确。系统实物效果图**：（1分）

戴着耳机的人

AI 生成的内容可能不正确。白色的游戏机

AI 生成的内容可能不正确。白色的游戏机

AI 生成的内容可能不正确。

**五、思考题（4分）**

如果想让舵机更精确，可以怎样操作？

** 提高 PWM 分辨率  
将 SysCtlPWMClockSet(SYSCTL\_PWMDIV\_1) 改为更高时钟（如 80 MHz / 2），并使用 16-bit 计数，可将脉宽分辨率提高到 0.025 µs 级。**

** 使用闭环位置反馈  
在舵轴安装光电编码器/电位计，通过 ADC 读取实际角度并用 PID 校正 PWM 脉宽，消除舵机死区。**

** 减小负载惯量  
减轻云台重量或供电提升至 6 V，可以减少回差并提高响应速度。**

** 数字滤波  
对目标角度使用一阶低通或“五点平均”滤波，可有效抑制量化抖动。**

**六、实验总结（2分）**

本次综合实验把 外设驱动、实时中断 和 串口交互 有机融合，完整走通了“检测-决策-执行-反馈”闭环流程。

通过 GPIO 双沿中断 成功实现毫秒级入侵检测，对 Tiva 的端口锁定/上拉配置有了更深体会；

PWM0/1 模块 驱动舵机与 LED，验证了占空比-角度/亮度的线性映射；

UARTStdio 简化了调试输出，串口指令使系统更具可控性；

状态机思想让代码逻辑清晰、易扩展，为后续加入多传感器或联网功能打下基础。

实验中还发现舵机在 5 V 供电下电压波动会影响 LED 亮度，今后可采用隔离电源或加入大容量滤波电容。