

ECE 4220

Zachary Rump

2017-04-09

Prelab 6

- `int rt_request_irq (unsigned irq, int(*handler)(unsigned irq, void *cookie), void *cookie, int retmode)`
 - Installs function handler as a standard Linux interrupt service routine for IRQ level irq.
 - Parameters
 - * handler - pointer to handler function.
 - * name - name for the handler under `/proc/interrupts`.
 - * dev_id - pass to the interrupt handler like standard Linux irq request.
 - Returns: 0 on success; EINVAL, EBUSY on failure.
- `void rt_enable_irq (unsigned irq)`
 - Enable an IRQ source.
 - Use this function instead of Linux equivalent when using real time handlers.
- `void rt_disable_irq (unsigned irq)`
 - Disable an IRQ source.
 - Use this function instead of Linux equivalent when using real time handlers
- `int rt_release_irq (unsigned irq)`
 - Release the IRQ from the handler.
 - Complement to `rt_request_irq()`.
- ep9301 registers that need modified for PORTB Interrupts
 - GPIOBIntEn - set bit enables interrupt for corresponding pin, 0 disable.
 - GPIOBIntType1 - set bit is edge sensitive, cleared is level sensitive interrupt.
 - GPIOBIntType2 - set bit is rising edge/high level, cleared is falling edge/low level.
 - GPIOBEOI - End of Interrupt register - set the bit to clear the interrupt. Clearing bit has no effect.
 - GPIOBDB - Set bit to enable debouncing of interrupt signals.
 - Procedure
 1. Disable interrupt
 2. Set interrupt type
 3. Clear interrupt
 4. Enable interrupt
 - Source Page 525 ep9301 manual.

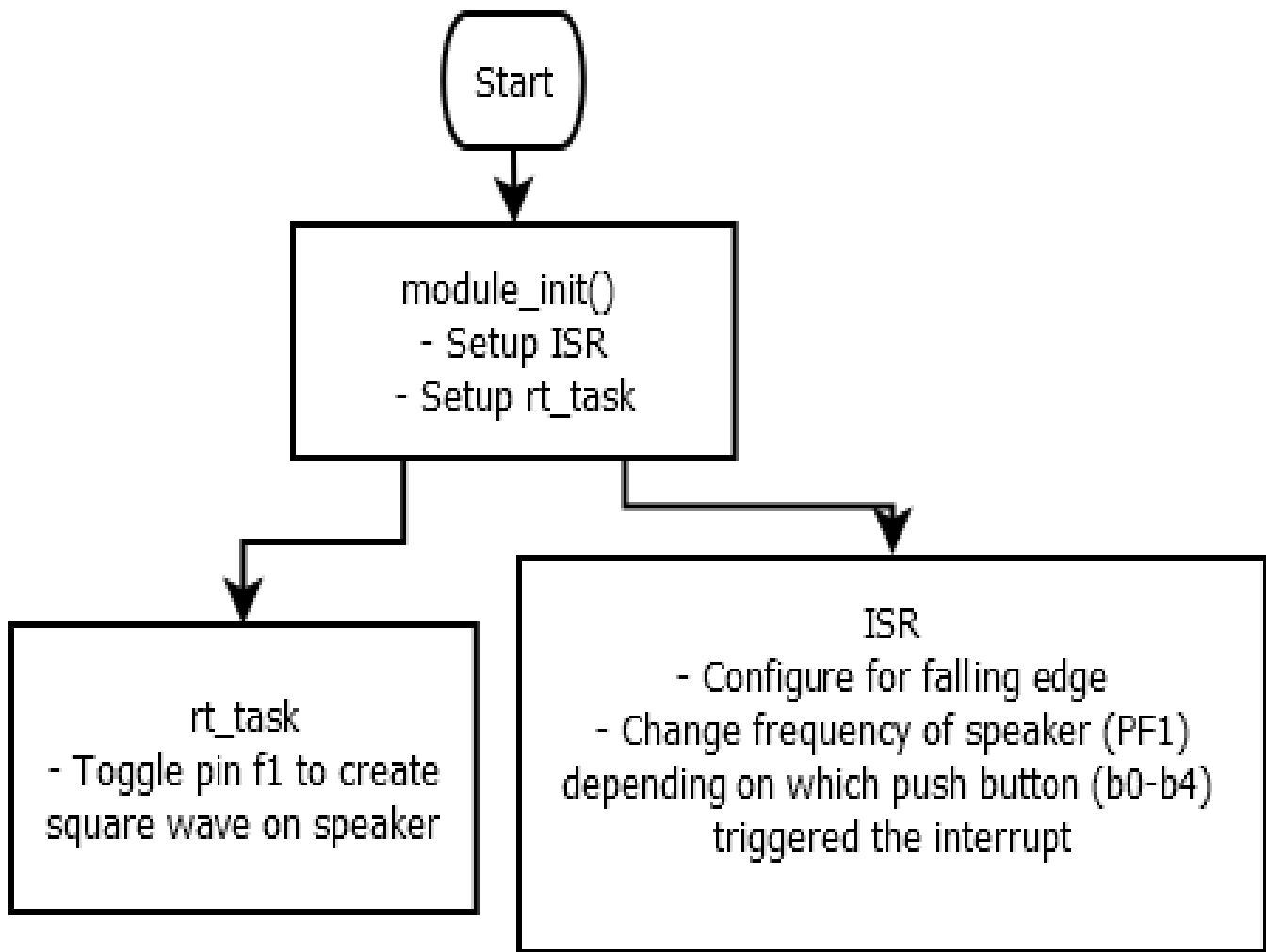


Figure 1. Flowchart for Lab 6 Part 1.

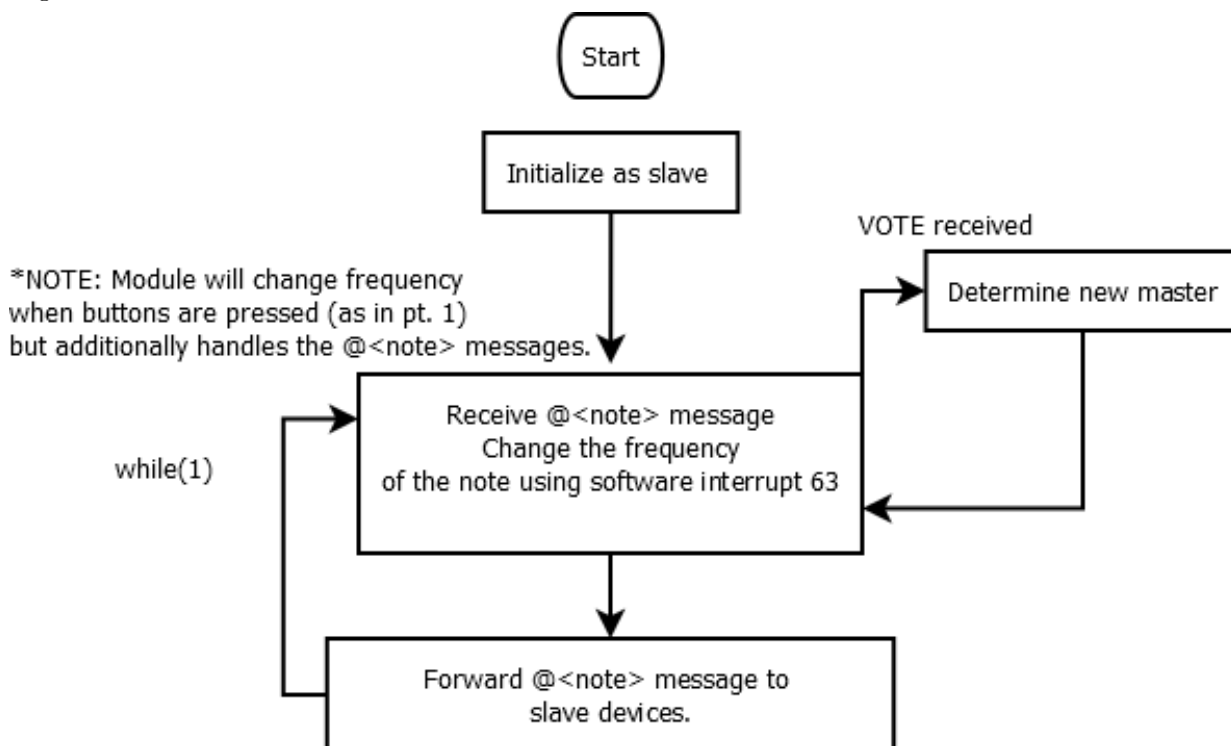


Figure 2. Flowchart for Lab 6 Part 2.