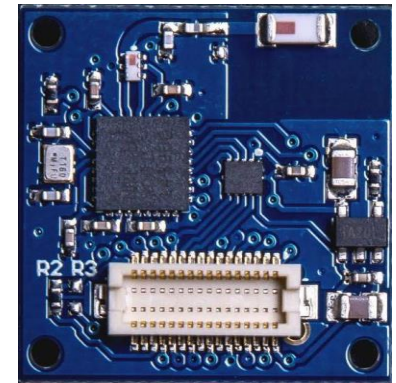
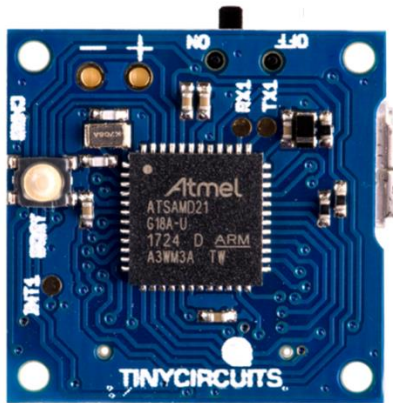


CSE190 Winter 2025

Lecture 24

Power Management



Wireless Embedded Systems

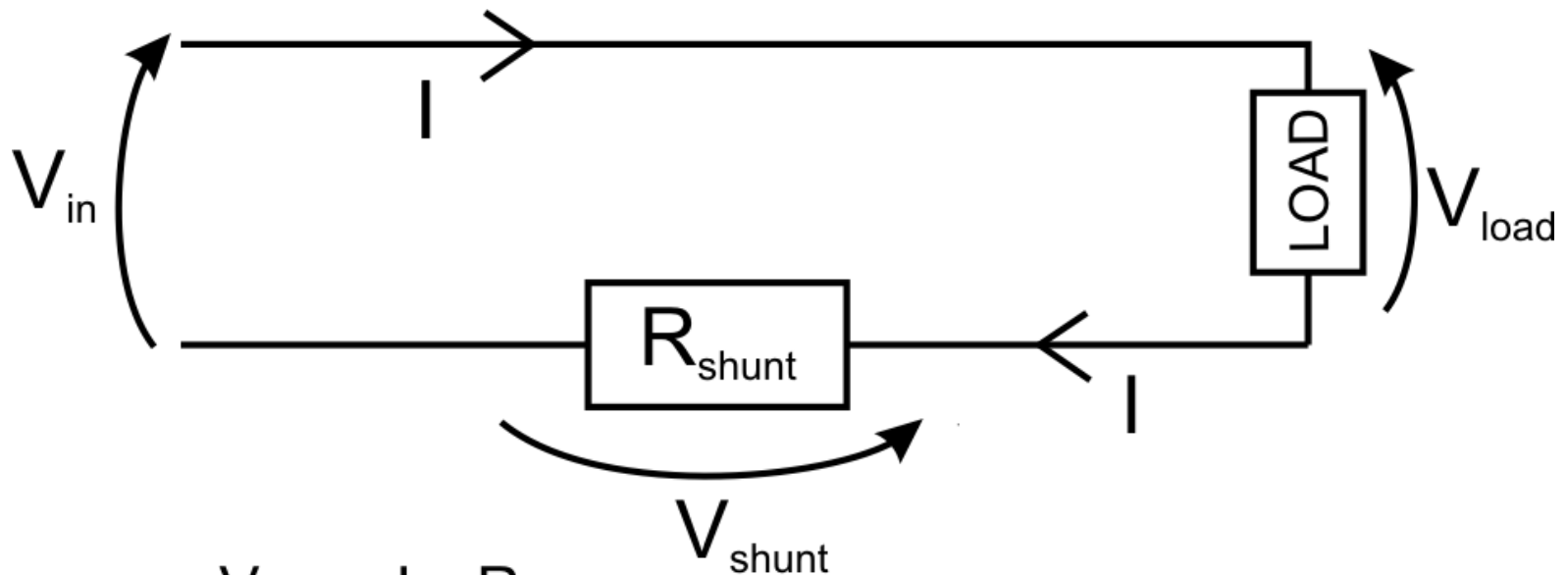
Aaron Schulman

How does your code let the CPU know it's ok to let it go to sleep?

- Special CPU instruction called "Wait for interrupt" or WFI
 - `__WFI()`
- Normally executed at the end of the `main()` loop
 - You are done handling all of the events that happened
- Need to set what CPU suspend mode you want before WFI

Stop 0 mode	Description
Mode entry	WFI (Wait for Interrupt) or WFE (Wait for Event) while: <ul style="list-style-type: none">– SLEEPDEEP bit is set in Cortex[®]-M4 System Control register– No interrupt (for WFI) or event (for WFE) is pending– LPMS = "000" in PWR_CR1
	On Return from ISR while: <ul style="list-style-type: none">– SLEEPDEEP bit is set in Cortex[®]-M4 System Control register– SLEEPONEXIT = 1– No interrupt is pending– LPMS = "000" in PWR_CR1

How do you measure the power of your embedded system?



$$V_{shunt} = I \times R_{shunt}$$

$$P = I^2 \times V_{load}$$

Insert a “shunt” resistor and measure two voltages: V_{load} and V_{shunt}

How much power is "low"

Coin cell battery is Volt, 220 milliampere hour



Main loop going into sleep

```
int main(void)
{
    while (1)
    {
        // TODO Your Main function

        // TODO Clear LPMS bits to set them to "000" (Stop mode)

        // Prepare to enter deep sleep mode (Stop mode)
        // Set the SLEEPDEEP bit in the System Control Register
        SCB_SCR |= SCR_SLEEPDEEP_Msk;

        // Execute the Wait-For-Interrupt instruction.
        // This puts the CPU into deep sleep mode until an interrupt occurs.
        __asm volatile ("wfi"); // same as __WFI();

        // After waking up, clear the SLEEPDEEP bit if you plan to return to a lighter sleep mode
        SCB_SCR &= ~SCR_SLEEPDEEP_Msk;

        // Optionally reconfigure clocks or perform wake-up tasks here
    }
}
```

Putting BLE into Standby Mode

When you are in non-discoverable mode then you can suspend the BLE radio:

```
standbyBLE();
```