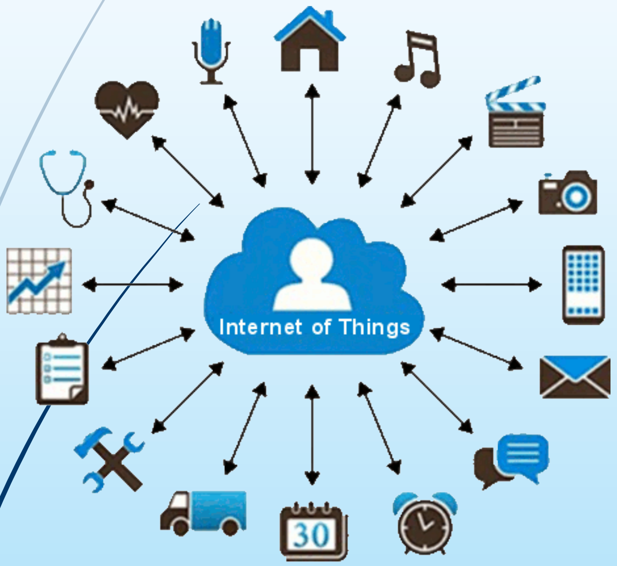
Practical Design Methodology



- Low-Power Design

Low-Power Modes

- There are FIVE Low Power modes on the STM32LO series
  - We are going to consider only FOUR
- Sleep Mode
  - Core is stopped, all peripherals keep running
  - Easy to exit and can be used in almost any situation
- Low-power Sleep Mode
  - Core is stopped and voltage regulator is placed into low power mode
- (Deep-sleep) Stop Mode
  - Core is stopped, clock oscillators run in a limited capacity
- (Deep-sleep) Standby Mode
  - Core is stopped, almost no clock oscillator and peripheral running

Low-Power Mode Configuration

- System Control Register (SCR)
  - DEEPSLEEP (bit 1)
  - SLEEPONEXIT (bit 0)
- DEEPSLEEP:
  - If set allows entry into a Deep sleep state, as opposed to the regular Sleep state
- SLEEPONEXIT
  - Applications tha only need to be awake to service interrupts
  - If set put processor into low-power mode after an exception return, before the program resumes execution

Low-Power Mode Configuration

- Power Control Register (PWR_CR)
  - LPSDSR (bit 0)
  - PDDS (bit 1)
  - CWUF (bit 2)

- LPSDSR (Low-Power Sleep-Deep/Sleep/low-power Run)
  - If set places the voltage regulator into a low-power state
- PDDS (Power Down DeepSleep)
  - If cleared the STOP mode is chosen over the STANDBY
- CWUF (Clear Wake-Up Flag)
  - If set clears the Wake-Up Flag

Low-Power Mode Configuration

- FLASH Access Control Register (FLASH_ACR)
  - SLEEP_PD (bit 1)

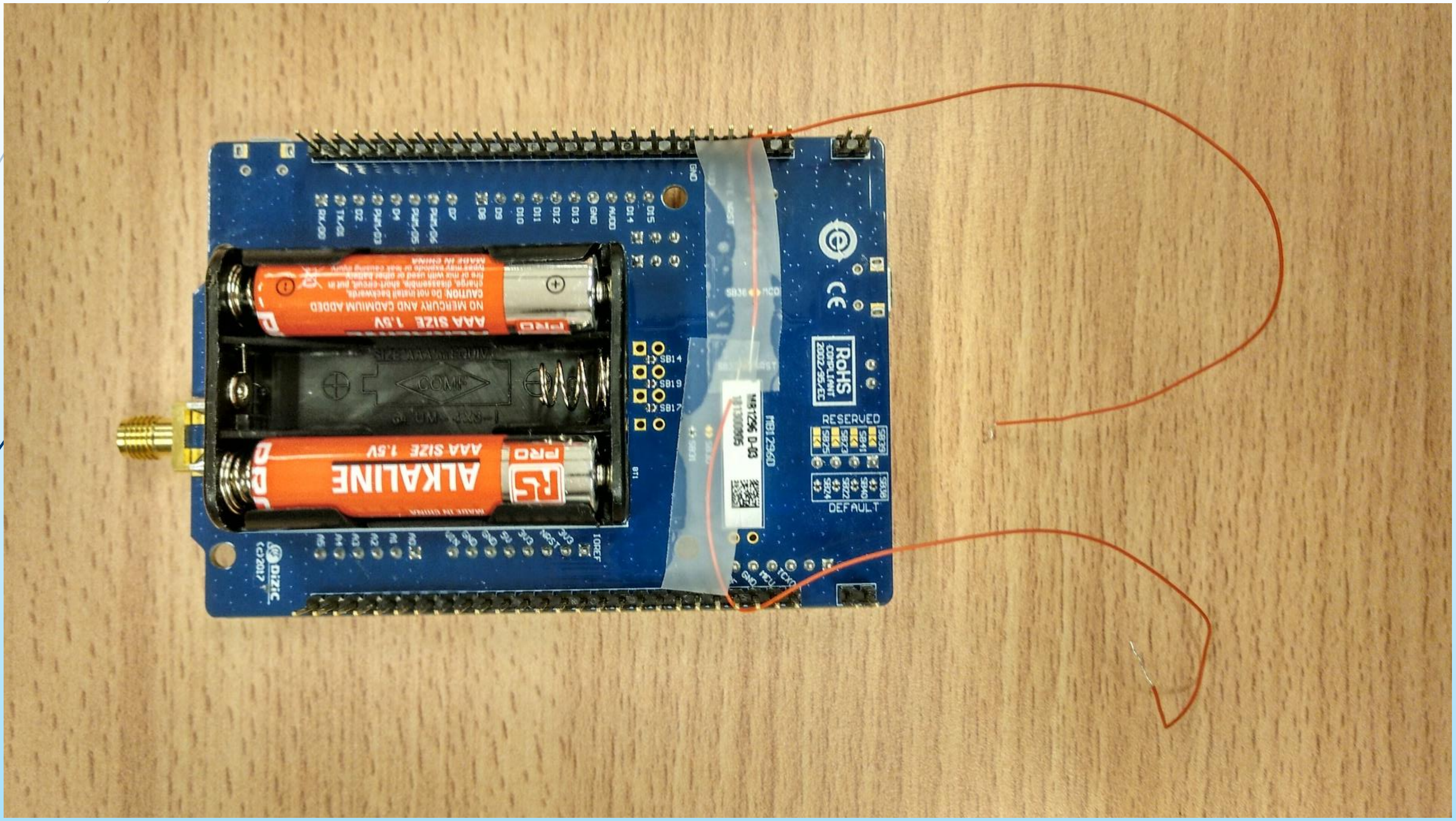
- SLEEP_PD ()
  - If set places the FLASH memory in power-down mode when the device enters either Sleep mode or Low-power sleep mode.

Enter Low-Power Mode

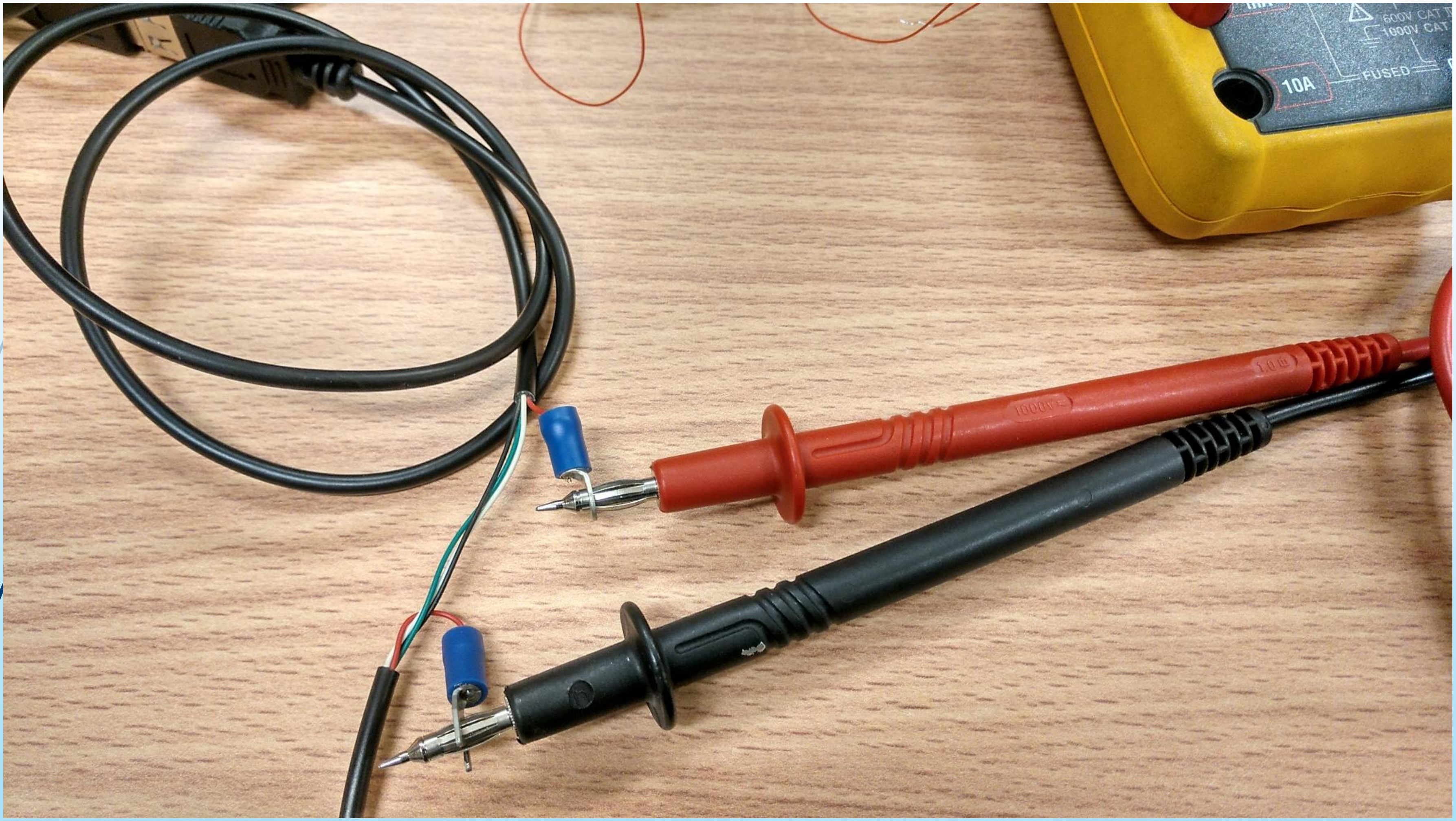
- Typical way …
  - Wait for Interrupt instruction
  - … although there are others (out of scope)

- MBED provides the function
  - __WFI()
  - To easily Access the WFI instruction when developing a C application

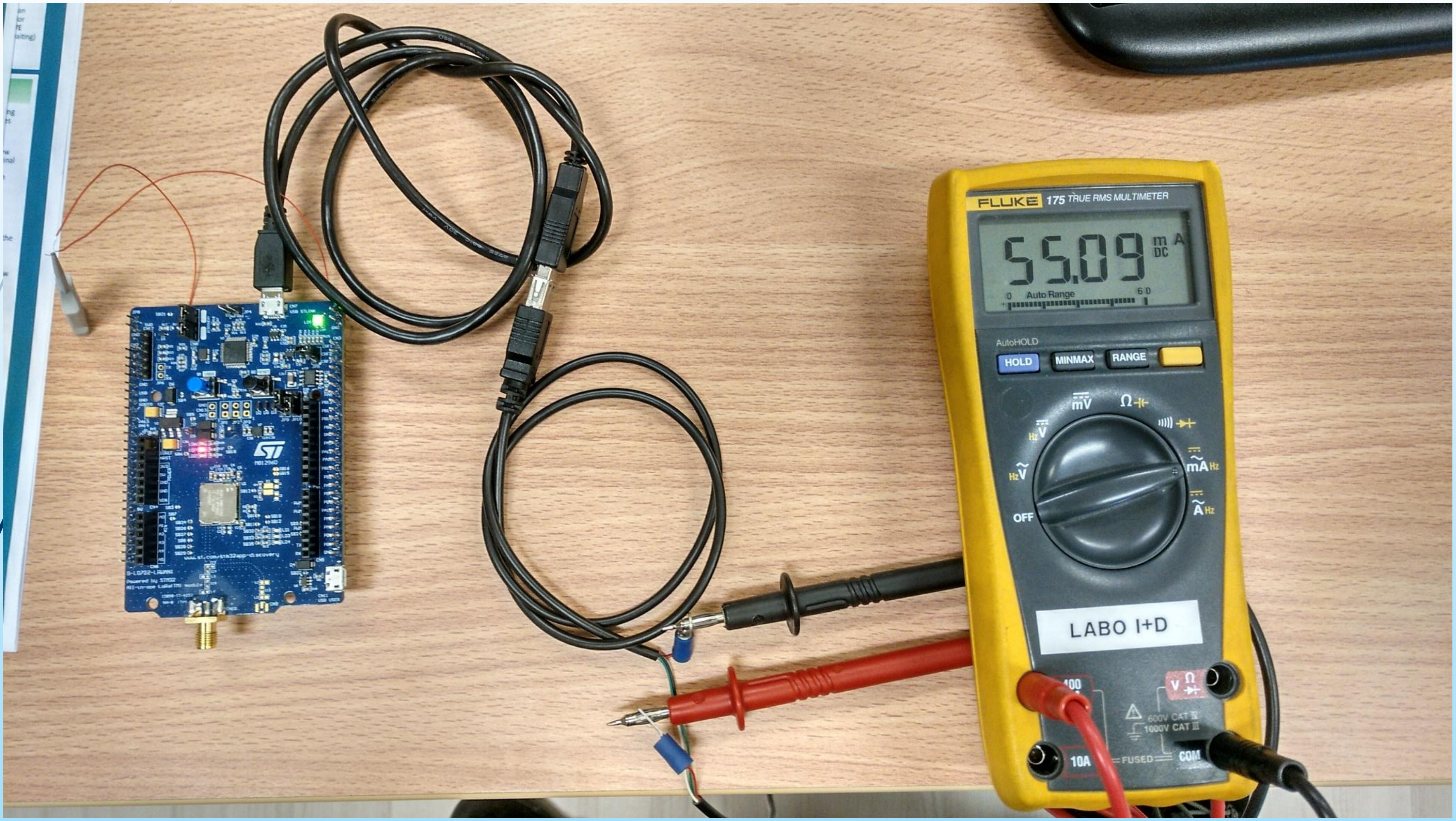
Power supplying by batteries



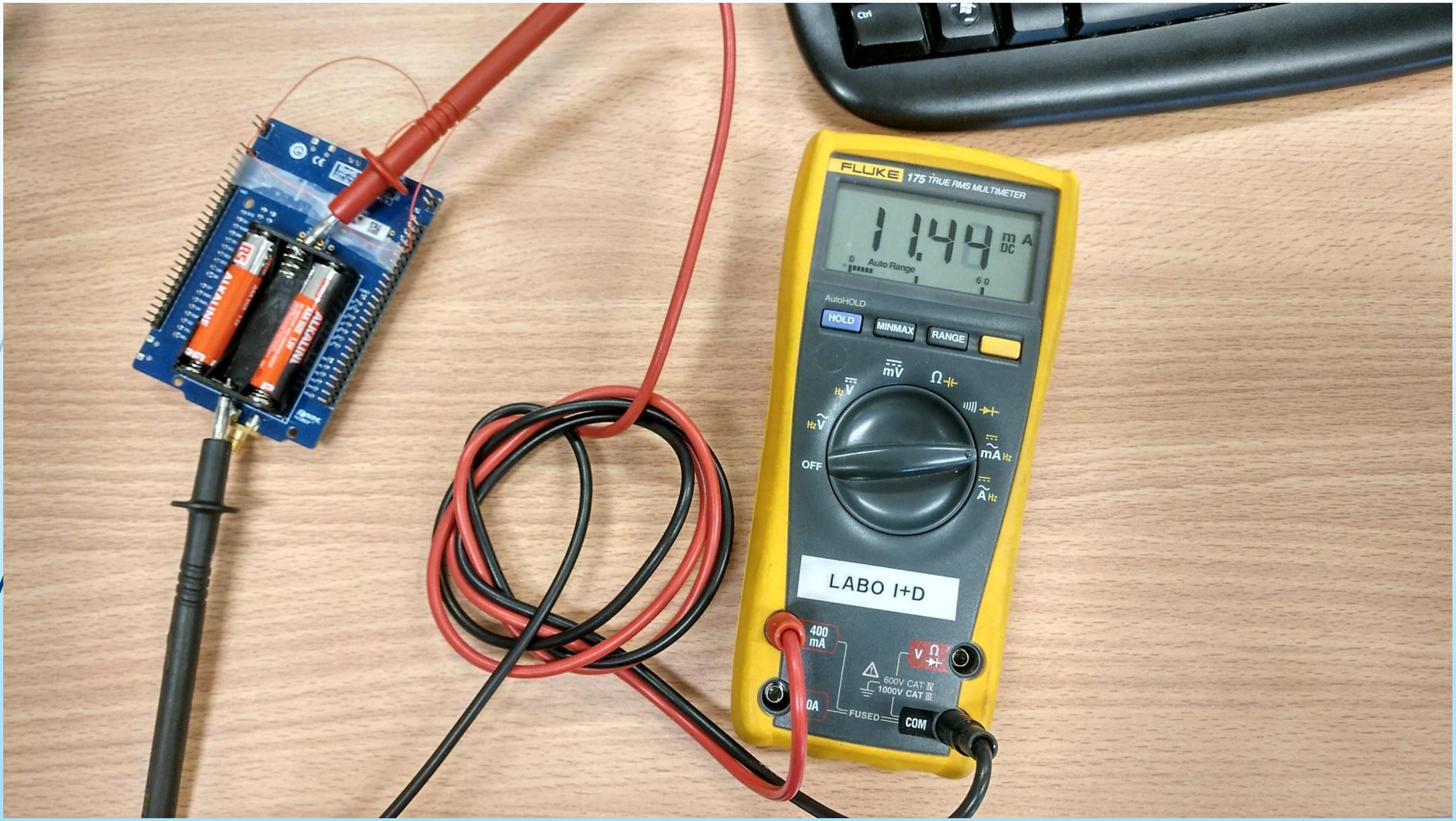
Measuring the power consumption



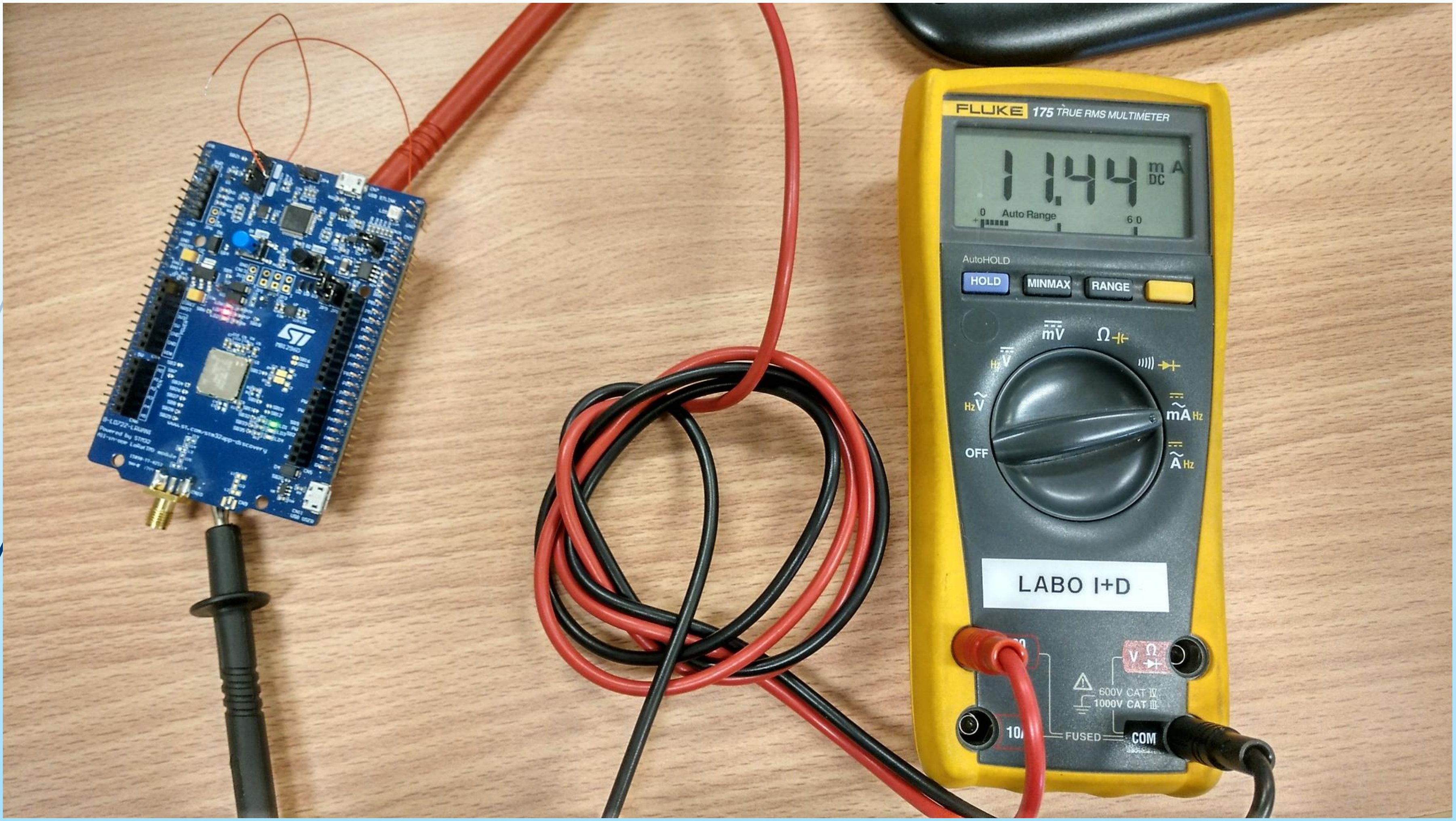
Test bench II: power-supplied by USB



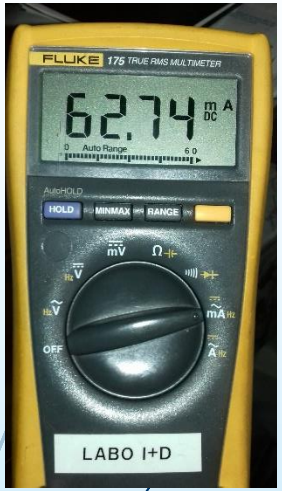
Test bench II: power-supplied by batteries (A-side)



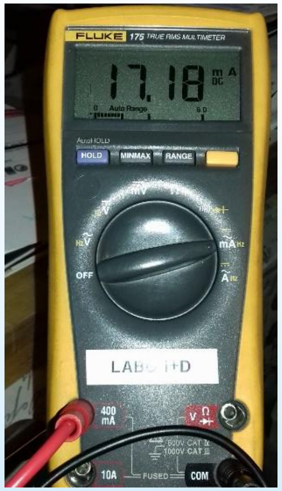
Test bench II: power-supplied by batteries (B-side)



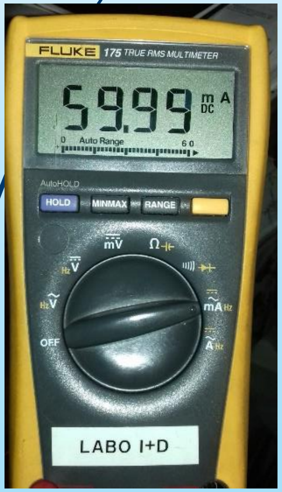
Using WAIT instruction



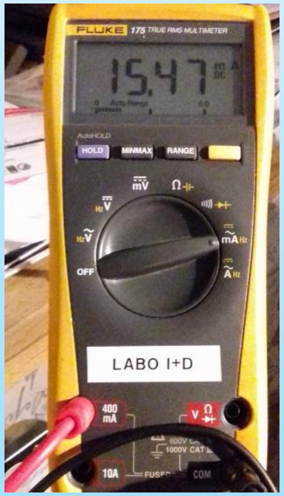
- ✓ USB
- ✓ LED ON



- ✓ Batteries
- ✓ LED ON



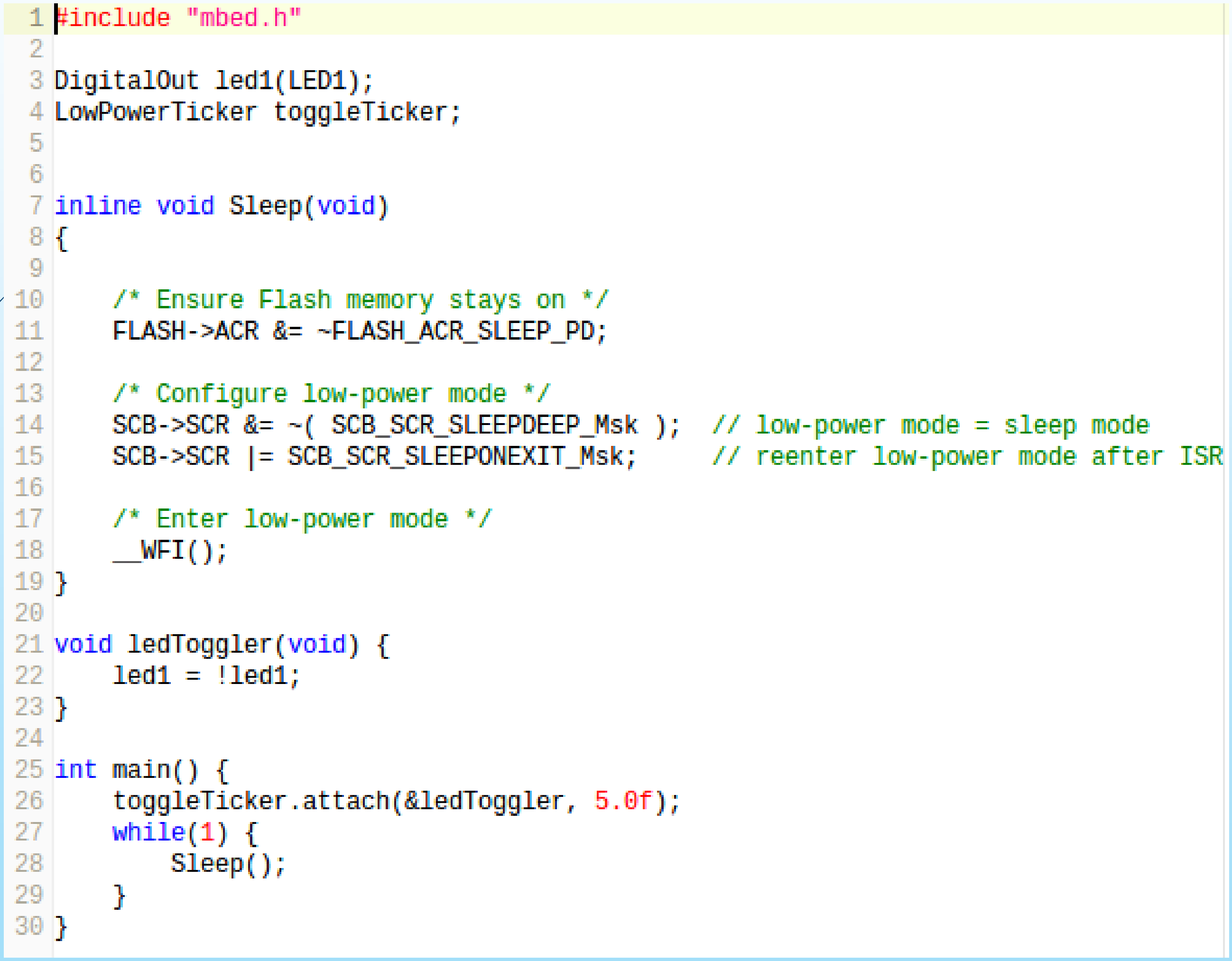
- ✓ USB
- ✓ LED OFF



- ✓ Batteries
- ✓ LED OFF

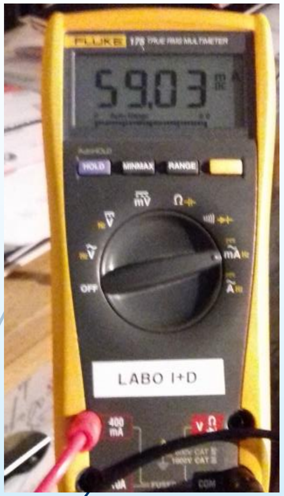
SLEEP Mode

Source Code

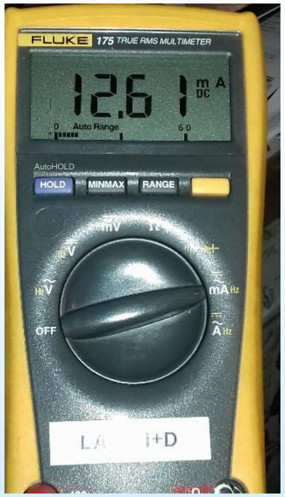


```
#include "mbed.h"

DigitalOut led1(LED1);
LowPowerTicker toggleTicker;


inline void Sleep(void)
{

    /* Ensure Flash memory stays on */
    FLASH->ACR &= ~FLASH_ACR_SLEEP_PD;

    /* Configure low-power mode */
    SCB->SCR &= ~( SCB_SCR_SLEEPDEEP_Msk );  // low-power mode = sleep mode
    SCB->SCR |= SCB_SCR_SLEEPONEXIT_Msk;     // reenter low-power mode after ISR

    /* Enter low-power mode */
    __WFI();
}

void ledToggler(void) {
    led1 = !led1;
}

int main() {
    toggleTicker.attach(&ledToggler, 5.0f);
    while(1) {
        Sleep();
    }
}
```

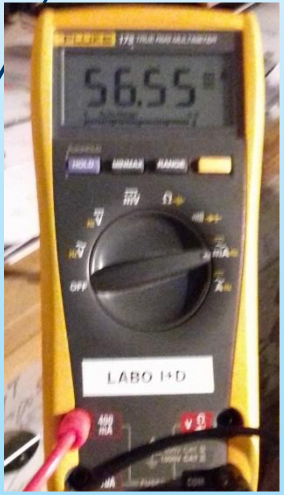
Using SLEEP mode



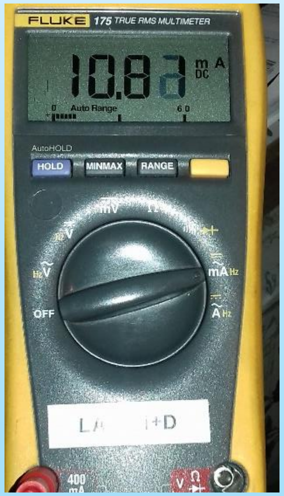
- ✓ USB
- ✓ LED ON



- ✓ Batteries
- ✓ LED ON



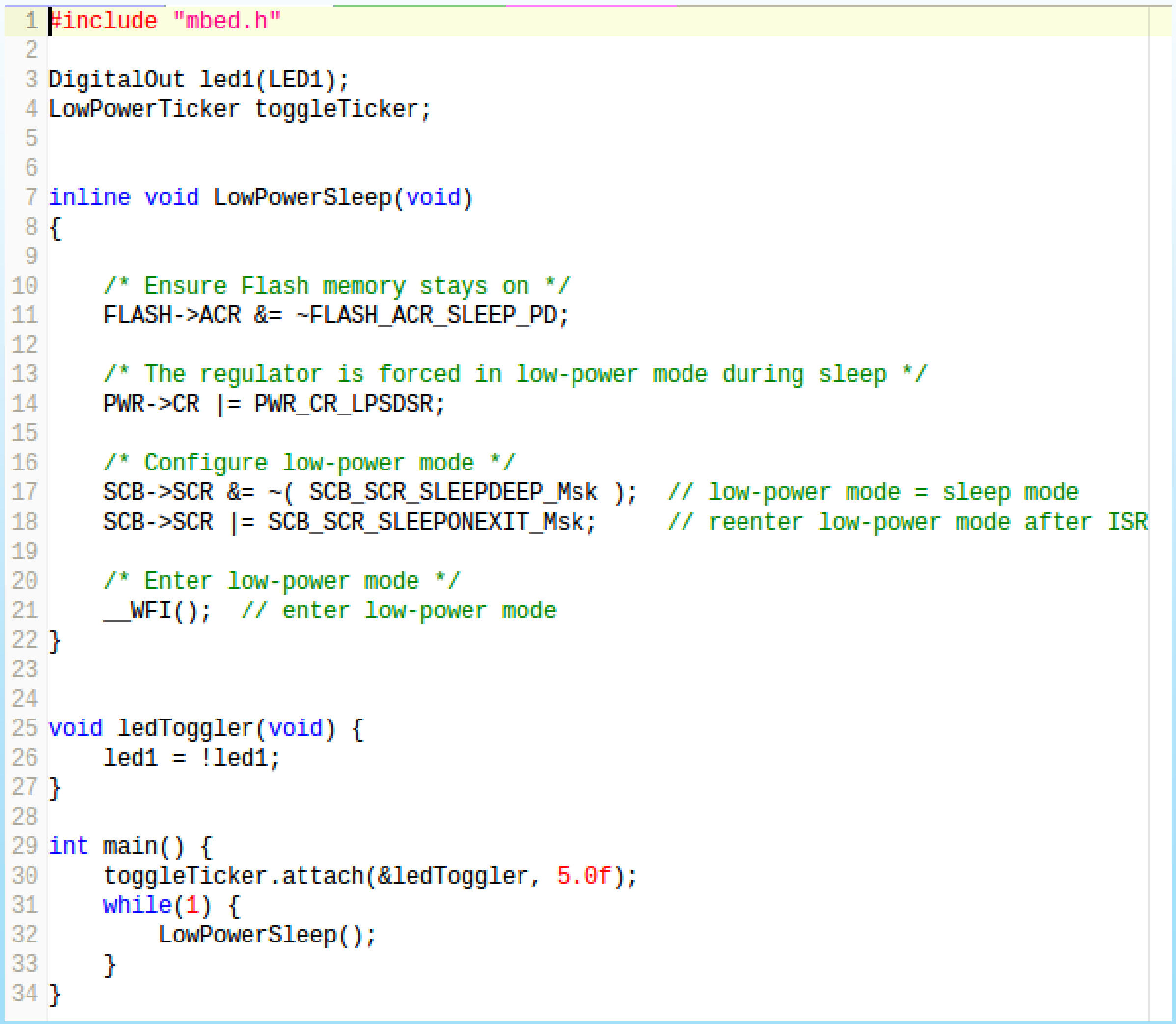
- ✓ USB
- ✓ LED OFF



- ✓ Batteries
- ✓ LED OFF

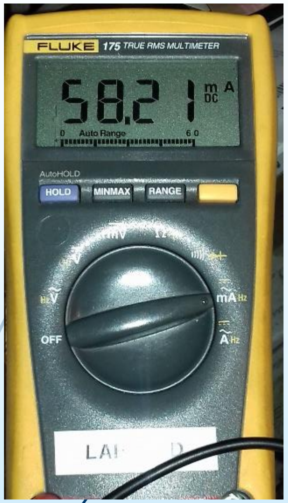
LOW POWER SLEEP Mode

Source Code

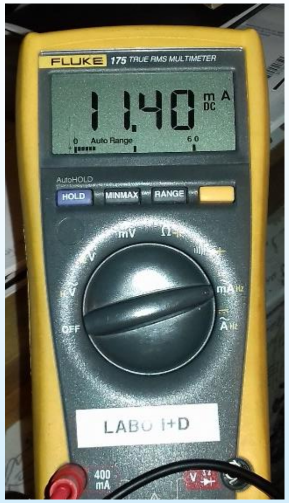


```
#include "mbed.h"

DigitalOut led1(LED1);
LowPowerTicker toggleTicker;


inline void LowPowerSleep(void)
{

    /* Ensure Flash memory stays on */
    FLASH->ACR &= ~FLASH_ACR_SLEEP_PD;

    /* The regulator is forced in low-power mode during sleep */
    PWR->CR |= PWR_CR_LPSDSR;

    /* Configure low-power mode */
    SCB->SCR &= ~( SCB_SCR_SLEEPDEEP_Msk );  // low-power mode = sleep mode
    SCB->SCR |= SCB_SCR_SLEEPONEXIT_Msk;     // reenter low-power mode after ISR

    /* Enter low-power mode */
    __WFI();  // enter low-power mode
}


void ledToggler(void) {
    led1 = !led1;
}

int main() {
    toggleTicker.attach(&ledToggler, 5.0f);
    while(1) {
        LowPowerSleep();
    }
}
```

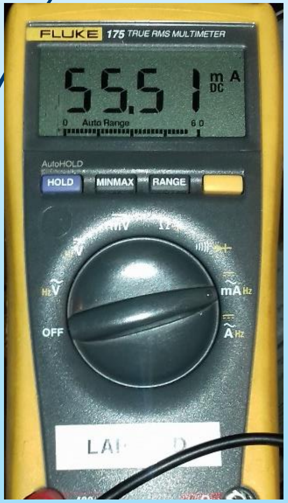
Using LOWPOWER SLEEP mode



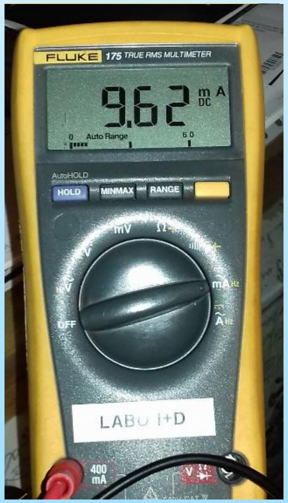
- ✓ USB
- ✓ LED ON



- ✓ Batteries
- ✓ LED ON



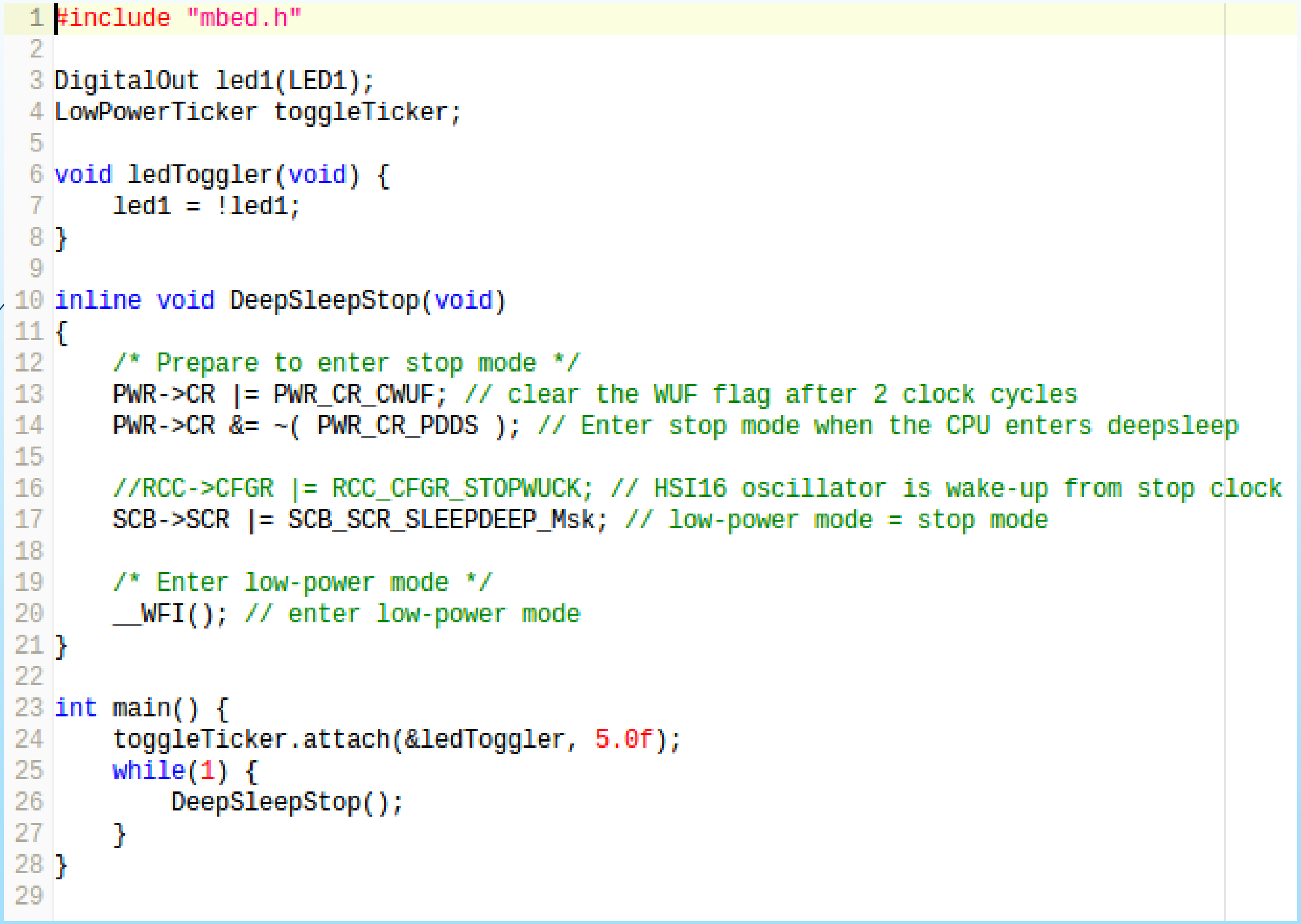
- ✓ USB
- ✓ LED OFF



- ✓ Batteries
- ✓ LED OFF

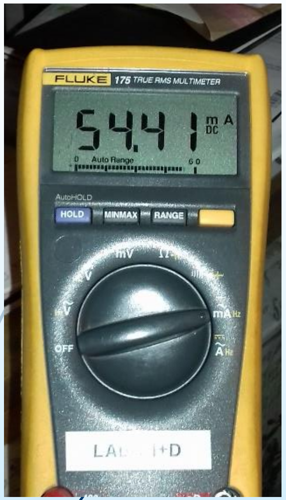
DEEP SLEEP STOP Mode

Source Code

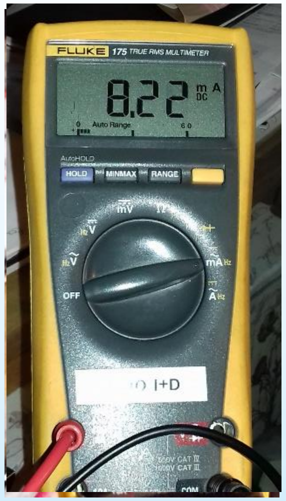


```
#include "mbed.h"

DigitalOut led1(LED1);
LowPowerTicker toggleTicker;

void ledToggler(void) {
    led1 = !led1;
}

inline void DeepSleepStop(void)
{
    /* Prepare to enter stop mode */
    PWR->CR |= PWR_CR_CWUF; // clear the WUF flag after 2 clock cycles
    PWR->CR &= ~( PWR_CR_PDDS ); // Enter stop mode when the CPU enters deepsleep

    //RCC->CFGR |= RCC_CFGR_STOPWUCK; // HSI16 oscillator is wake-up from stop clock
    SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk; // low-power mode = stop mode

    /* Enter low-power mode */
    __WFI(); // enter low-power mode
}

int main() {
    toggleTicker.attach(&ledToggler, 5.0f);
    while(1) {
        DeepSleepStop();
    }
}

```

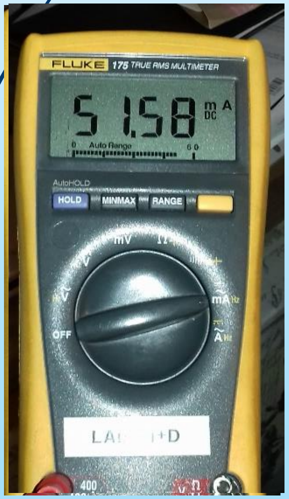
Using DEEP SLEEP STOP MODE



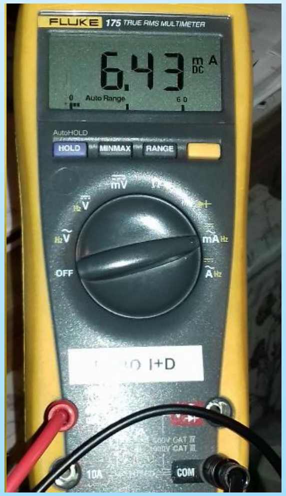
- ✓ USB
- ✓ LED ON



- ✓ Batteries
- ✓ LED ON



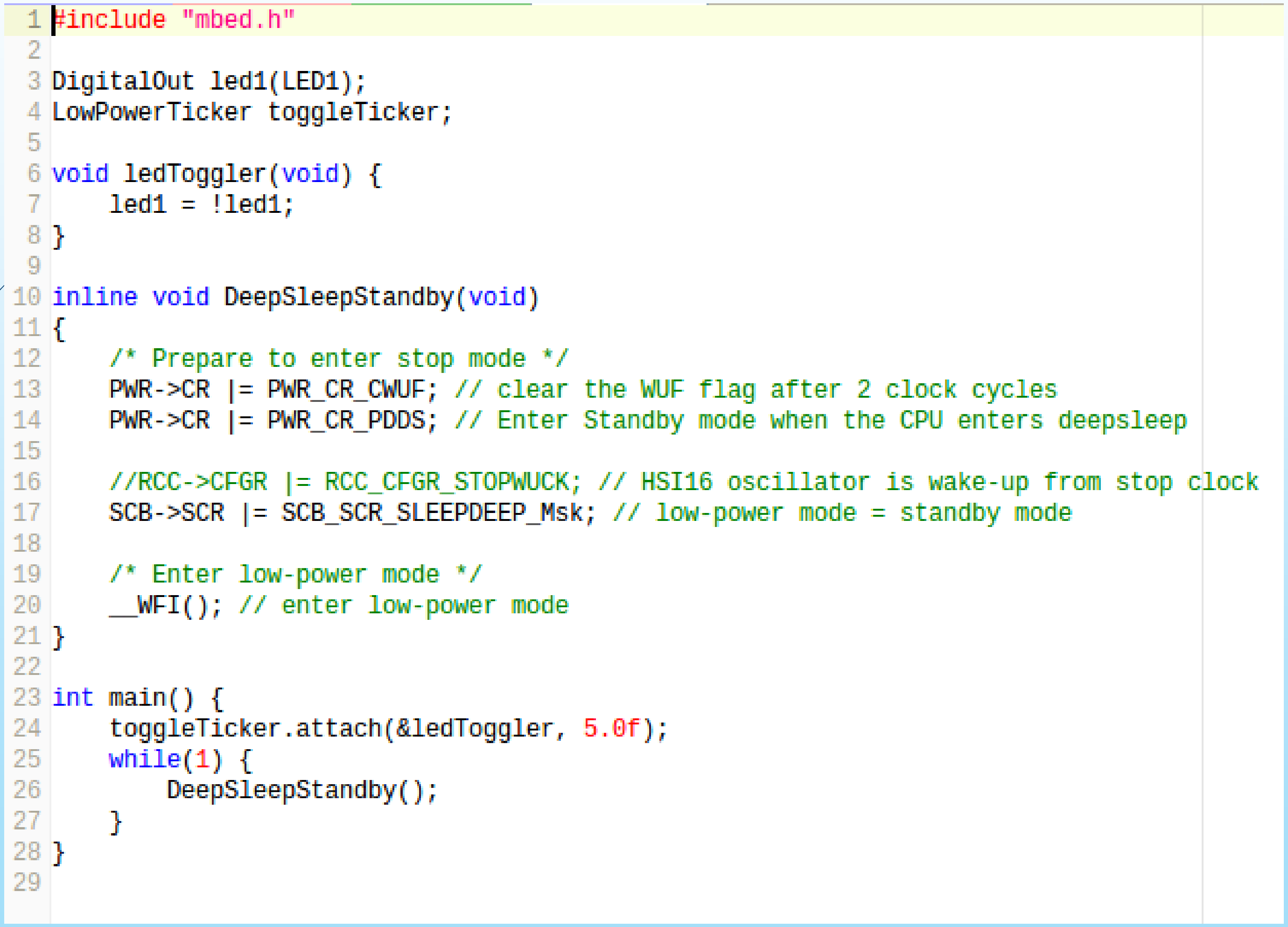
- ✓ USB
- ✓ LED OFF



- ✓ Batteries
- ✓ LED OFF

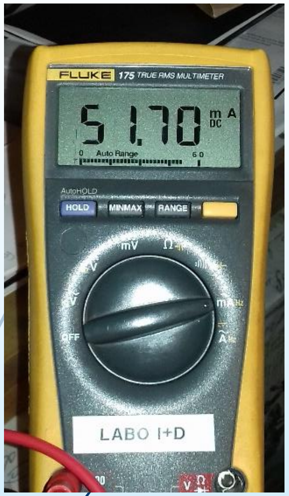
DEEP SLEEP STANDBY Mode

Source Code

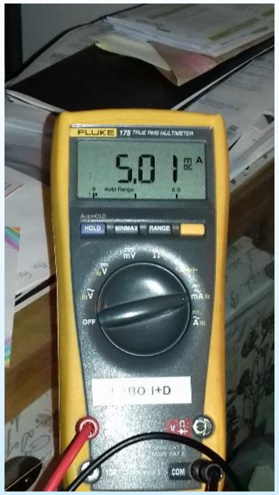


```
#include "mbed.h"

DigitalOut led1(LED1);
LowPowerTicker toggleTicker;

void ledToggler(void) {
    led1 = !led1;
}

inline void DeepSleepStandby(void)
{
    /* Prepare to enter stop mode */
    PWR->CR |= PWR_CR_CWUF; // clear the WUF flag after 2 clock cycles
    PWR->CR |= PWR_CR_PDDS; // Enter Standby mode when the CPU enters deepsleep

    //RCC->CFGR |= RCC_CFGR_STOPWUCK; // HSI16 oscillator is wake-up from stop clock
    SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk; // low-power mode = standby mode

    /* Enter low-power mode */
    __WFI(); // enter low-power mode
}

int main() {
    toggleTicker.attach(&ledToggler, 5.0f);
    while(1) {
        DeepSleepStandby();
    }
}

```

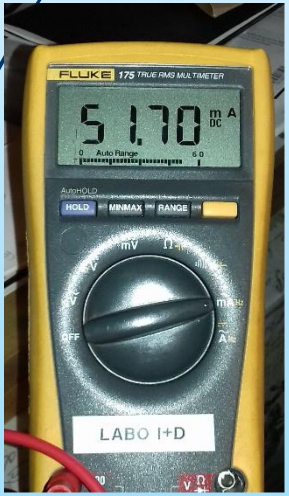
Using DEEP SLEEP STANDBY MODE



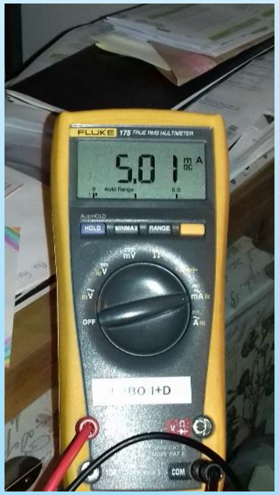
- ✓ USB
- ✓ LED ON



- ✓ Batteries
- ✓ LED ON



- ✓ USB
- ✓ LED OFF



- ✓ Batteries
- ✓ LED OFF

Summary

MODE	LEDs	Consumption with USB (mA)	Consumption with batteries (mA)	% Savings regarding (1)	
WAIT	ON	**63.0 (1)**	17.2	-	72.7
	OFF	60.0	15.5	4.8	75.4
SLEEP	ON	59.0	12.6	6.3	80.0
	OFF	56.5	10.8	10.3	82.9
LOWPOWER SLEEP	ON	58.2	11.4	7.6	81.9
	OFF	55.5	9.6	11.9	84.8
DEEP SLEEP STOP MODE	ON	54.9	8.2	**12.9**	**87.0**
	OFF	52.1	6.4	17.3	89.8