

How to use a Photocell with RIOT

using an STM32 Nucleo-64 F401RE development board

Ioannis Chatzigiannakis

<https://github.com/ichatz/riotos-apps>

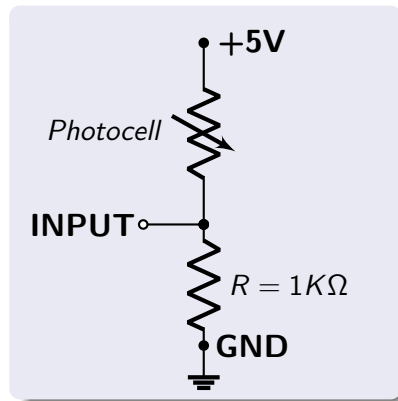


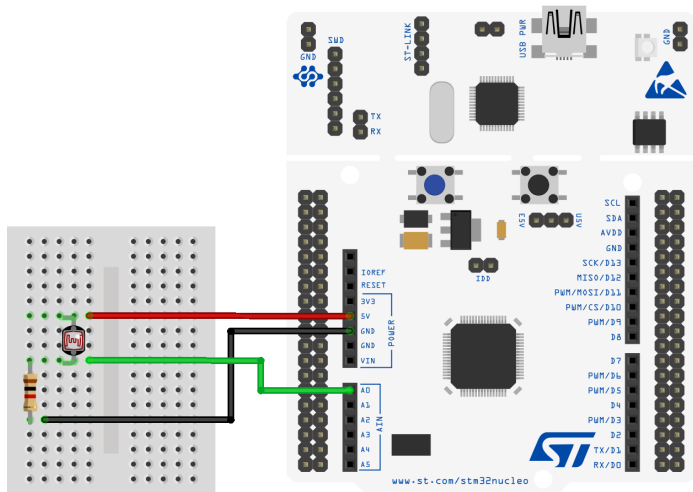
Measure the intensity of ambient light with an STM32 Nucleo-64 F401RE development board and the RIOT operating system.

Required hardware components:

- STM32 Nucleo-64 F401RE
- Breadboard
- $1K\Omega$ resistor
- Photocell (or Photoresistor, or Light-Dependent-Resistor LDR)
- 3 Male to male jumper wires

- The photocell is a light-controlled variable resistor:
 - When struck by light – resistance decreases.
 - In absence of light – resistance increases.
- Measure the variable resistor → measure intensity of ambient light.
- Measure a variable resistance:
 - Connect one end to power (+5V),
 - Connect the other to a pull-down resistor ($1K\Omega$) to ground (GND).
 - Connect the point INPUT to the analog input of the STM32 board.
- The pull-down resistor depends on:
 - Input voltage (5V or 3.3V or ...).
 - Intensity of ambient light.



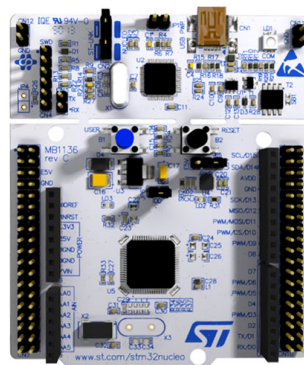


fritzing

The STM32 Nucleo-64 board provides an affordable and flexible way for users to try out new concepts and build prototypes by choosing from the various combinations of performance and power consumption features, provided by the STM32 microcontroller.

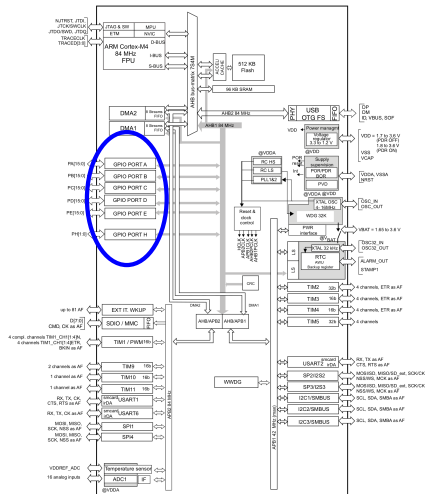
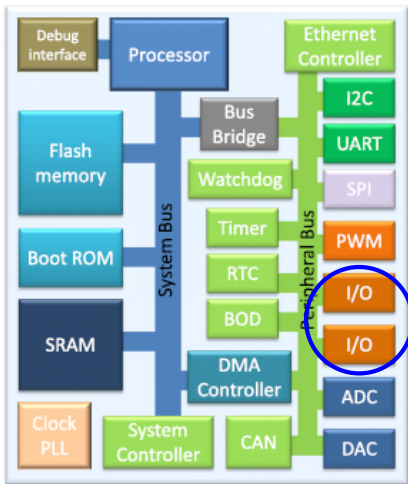
Highlights of the STM32F401RE MCU:

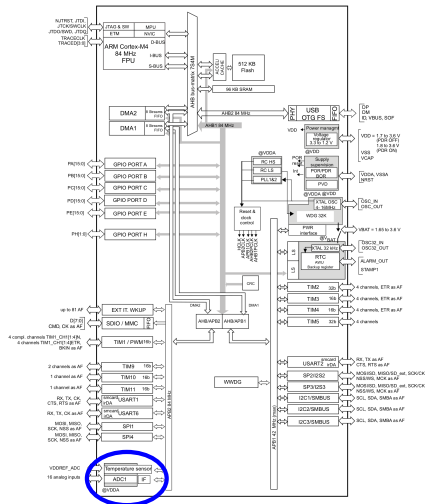
- 32-bit ARM Cortex-M4 processor
- 512KB ROM Flash memory
- 96KB SRAM data memory
- Up to 84Mhz operating frequency
- 42 μ A in sleep (stop mode) w/RTC



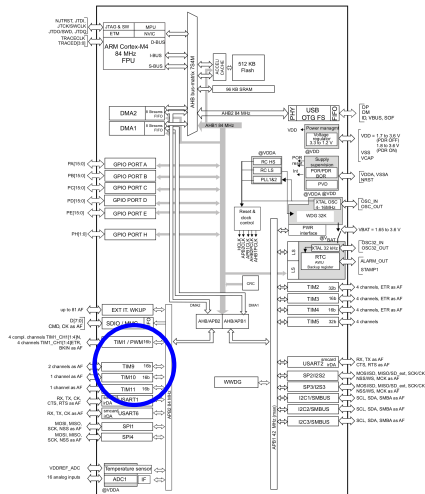
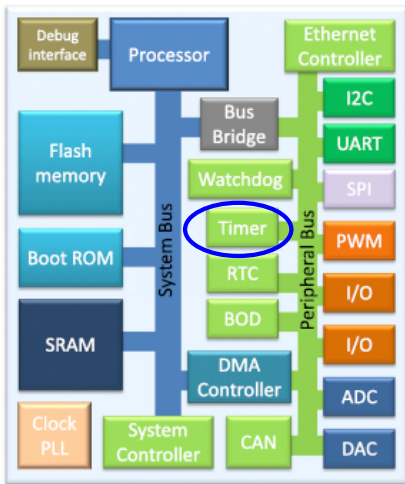


STM32F401RE Architecture

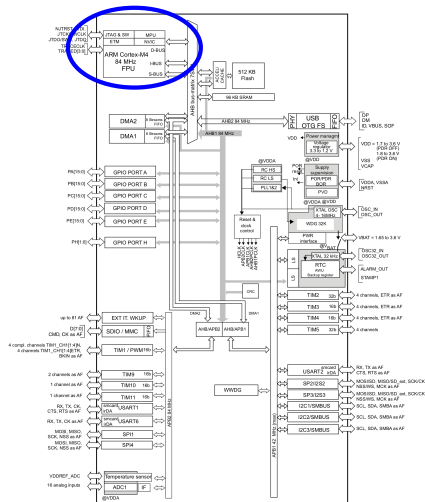
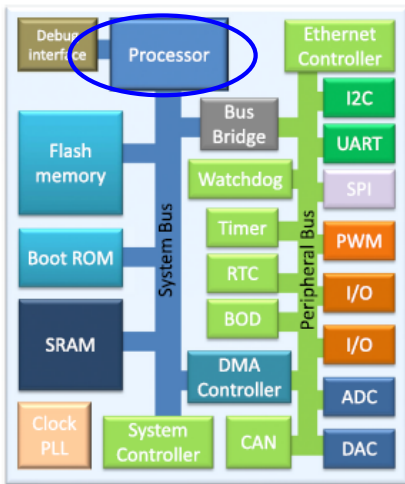




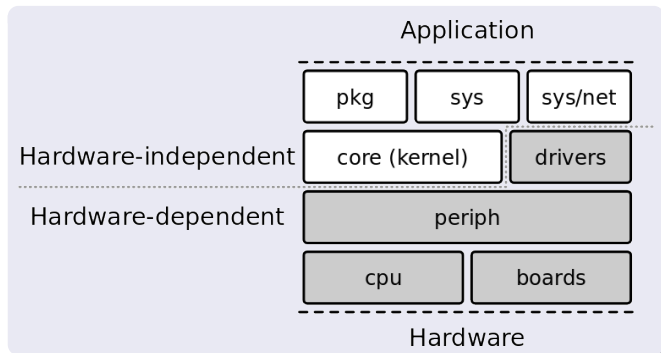
STM32F401RE Architecture



STM32F401RE Architecture



RIOT Architecture



**Application logic
+ RIOT Kernel + RIOT
Libraries**

- Generic API for peripherals (gpio, uart, spi, pwm, etc)
 - same API for all architectures
- CPU, Boards and Drivers: manufacturer specific



Hardware Independent elements

Makefile

```
# name of application
APPLICATION = photocell

# Path to the RIOT base directory:
RIOTBASE ?= $(CURDIR)/../../RIOT

# RIOT features
FEATURES_REQUIRED += periph_adc

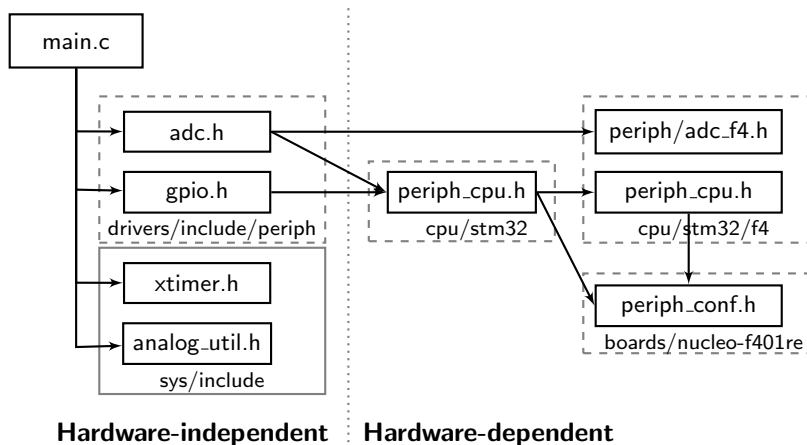
# Modules to include:
USEMODULE += analog_util
USEMODULE += xtimer
```

We specify the ADC as **required** feature.

We also specify that we want to use two **OS modules**.



Hardware Dependent elements



main.c

```
int main(void) {  
    if (adc_init(ADC_LINE(0)) < 0) {  
        printf("Initialization of ADC_LINE failed\n");  
        return 1;  
    }  
  
    int sample = adc_sample(ADC_LINE(0), ADC_RES_12BIT);  
    lux = adc_util_map(sample, ADC_RES_12BIT, 10, 100);  
}
```

- Our MCU has a single ADC with 12bit resolution.
- A conversion gives up to 4096 different values.
- Input range is from 0V ... 5V, thus each single value represents 1.2mV.
- We map each value to the lux range 10 ... 100.

main.c

```
int main(void) {
    xtimer_ticks32_t last = xtimer_now();

    while (1) {
        int sample = adc_sample(ADC_LINE(0), ADC_RES_12BIT);
        lux = adc_util_map(sample, ADC_RES_12BIT, 10, 100);

        xtimer_periodic_wakeup(&last, 100LU * US_PER_MS);
    }
}
```

- Module xtimer multiplexes hardware timers, with microseconds accuracy.
- System time resolution is 32bit.
- xtimer_periodic_wakeup() suspends the calling thread for 100ms.