Programmazione di sistemi multicore

Michele Martinelli
Michele.martinelli@uniroma1.it

Correzione esonero

Programma di oggi:

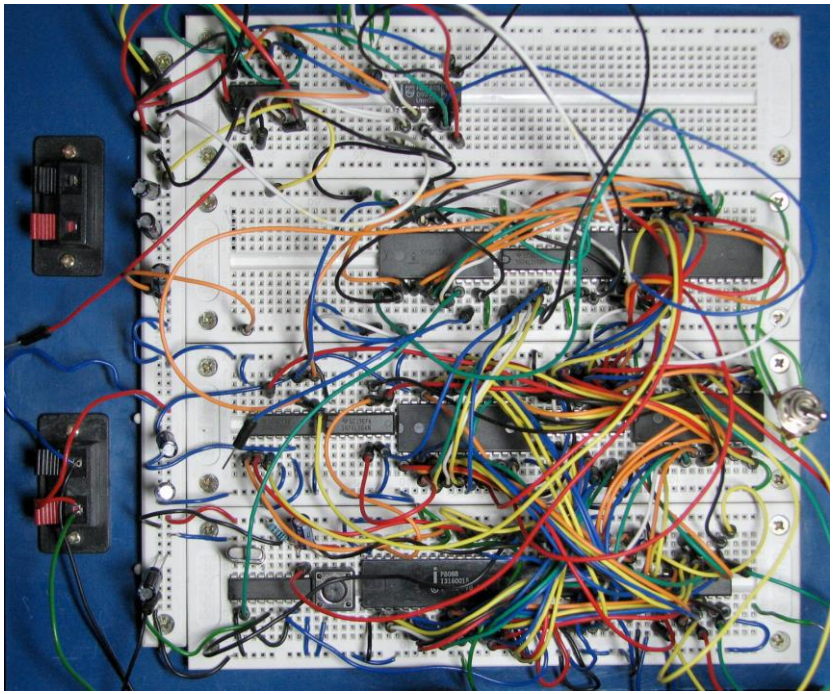
- Correzione esonero (i risultati saranno pubblicati a breve)
- Come si realizzano PCB (perché facciamo prototipi su breadboard? E perché serve Arduino?)
- Protocolli seriali di comunicazione (UART, SPI, I2C)
- Porte e registri
- Esempi e esercizi semplici su NUCLEO board (PWM con LED + FreeRTOS)

Programma di martedì prossimo

- Teoria FreeRTOS (timer, interrupt)
- Esercizi su NUCLEO board (FreeRTOS + motori DC, servomotori)
- Revisione compiti

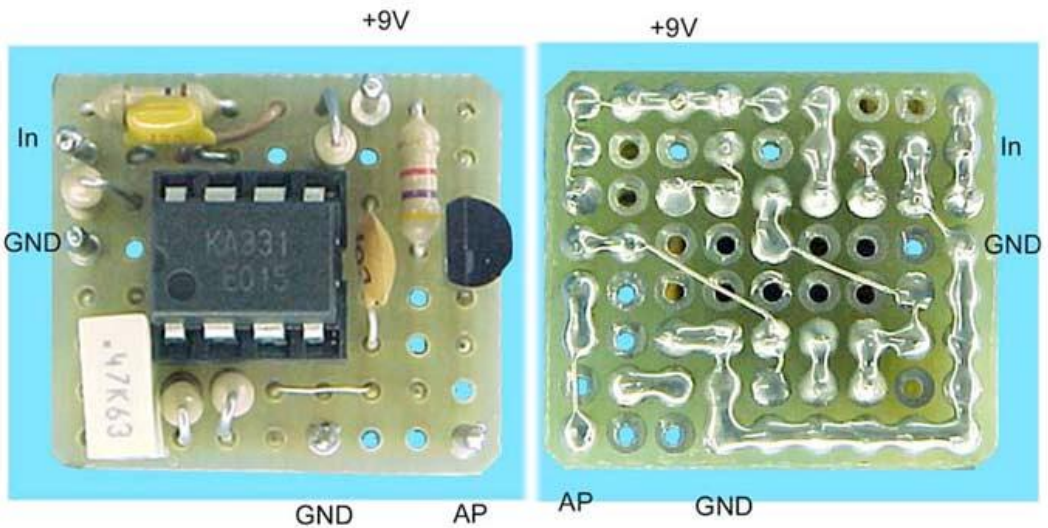
PER CHI VOLESSE APPROFONDIRE LA PARTE DI ARM

Realizzare PCB – fase 1



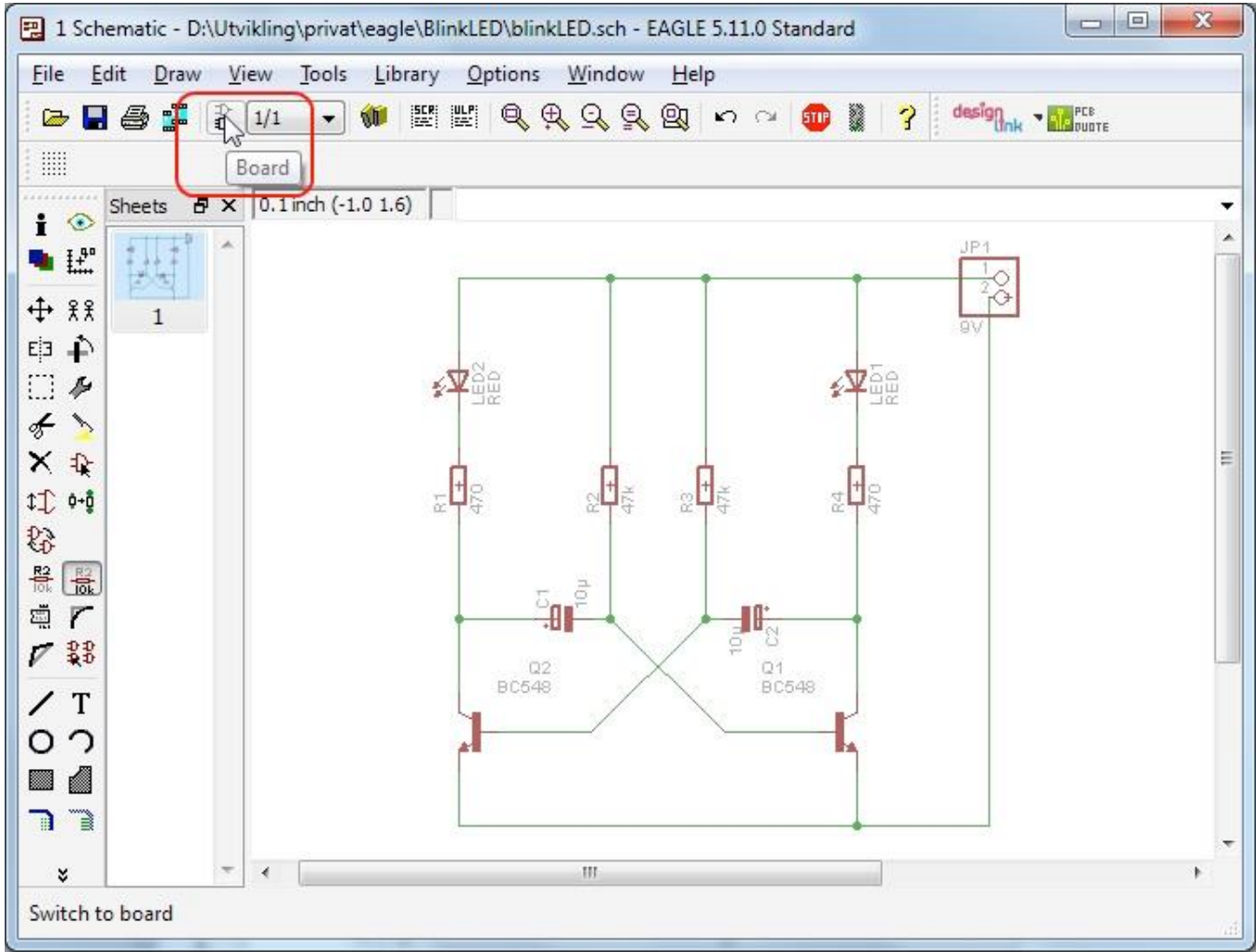
Progettare e testare il circuito su breadboard o millefori

Questo permette di sviluppare e testare software e hardware



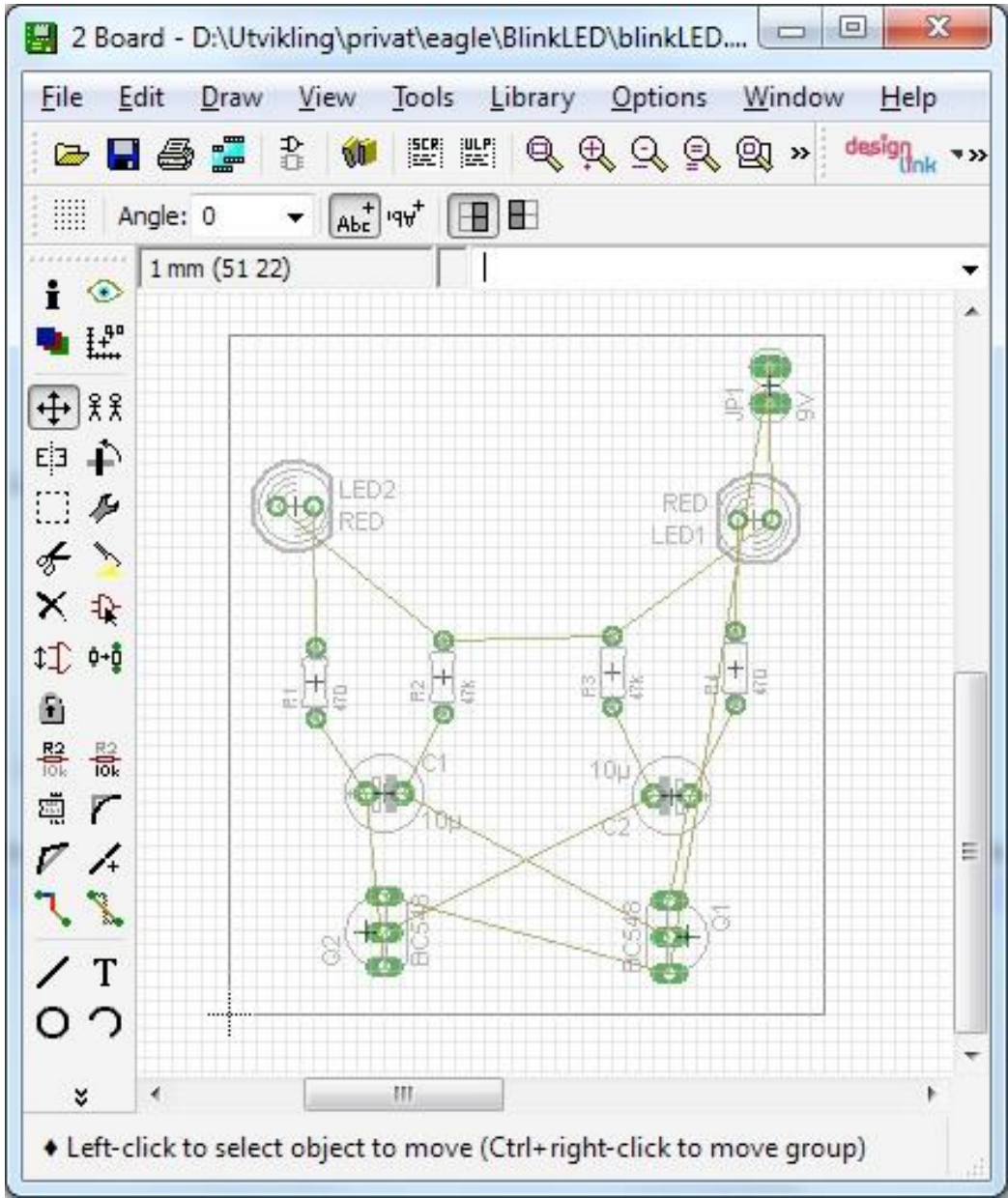
Realizzare PCB – fase 2

Disegnare lo schematico al cad



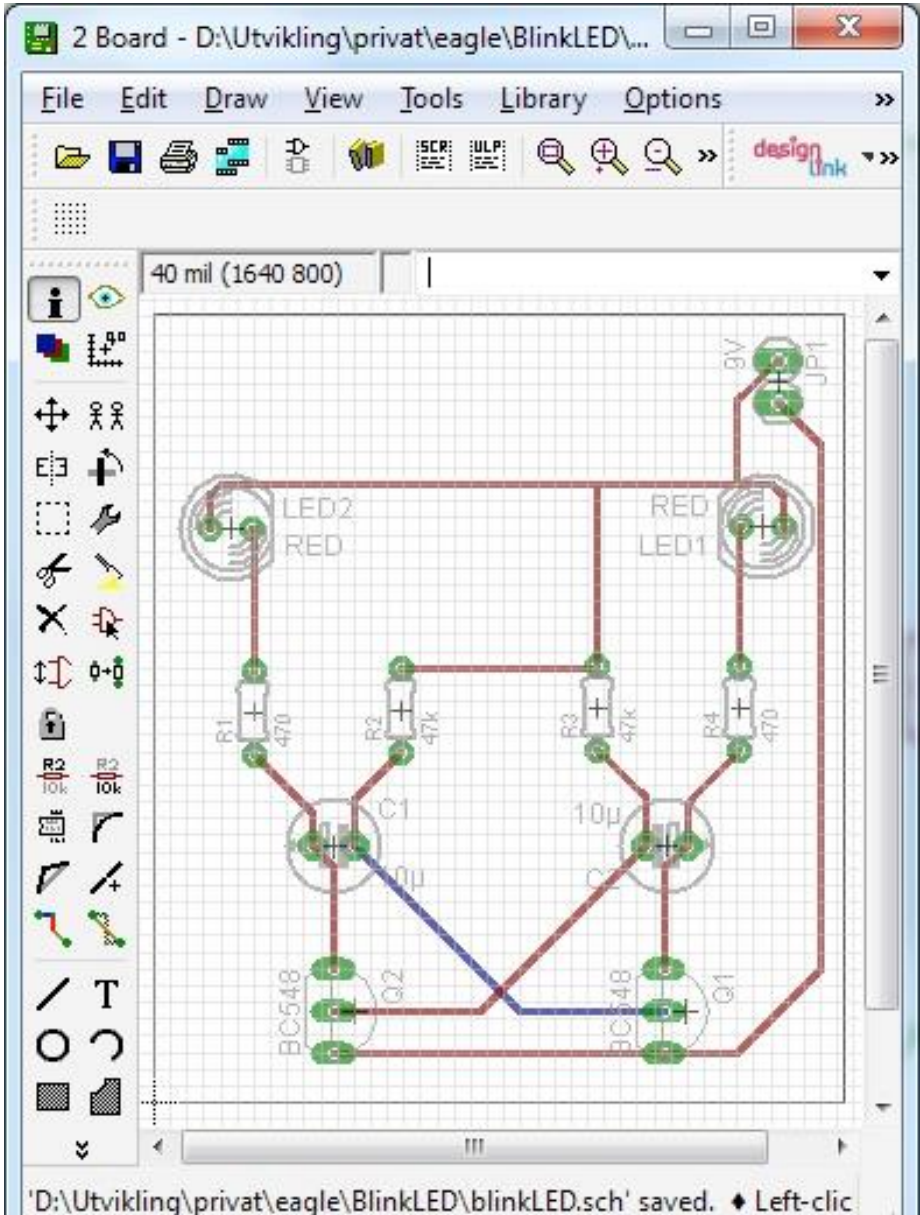
Realizzare PCB – fase 3

Disegnare la board al cad



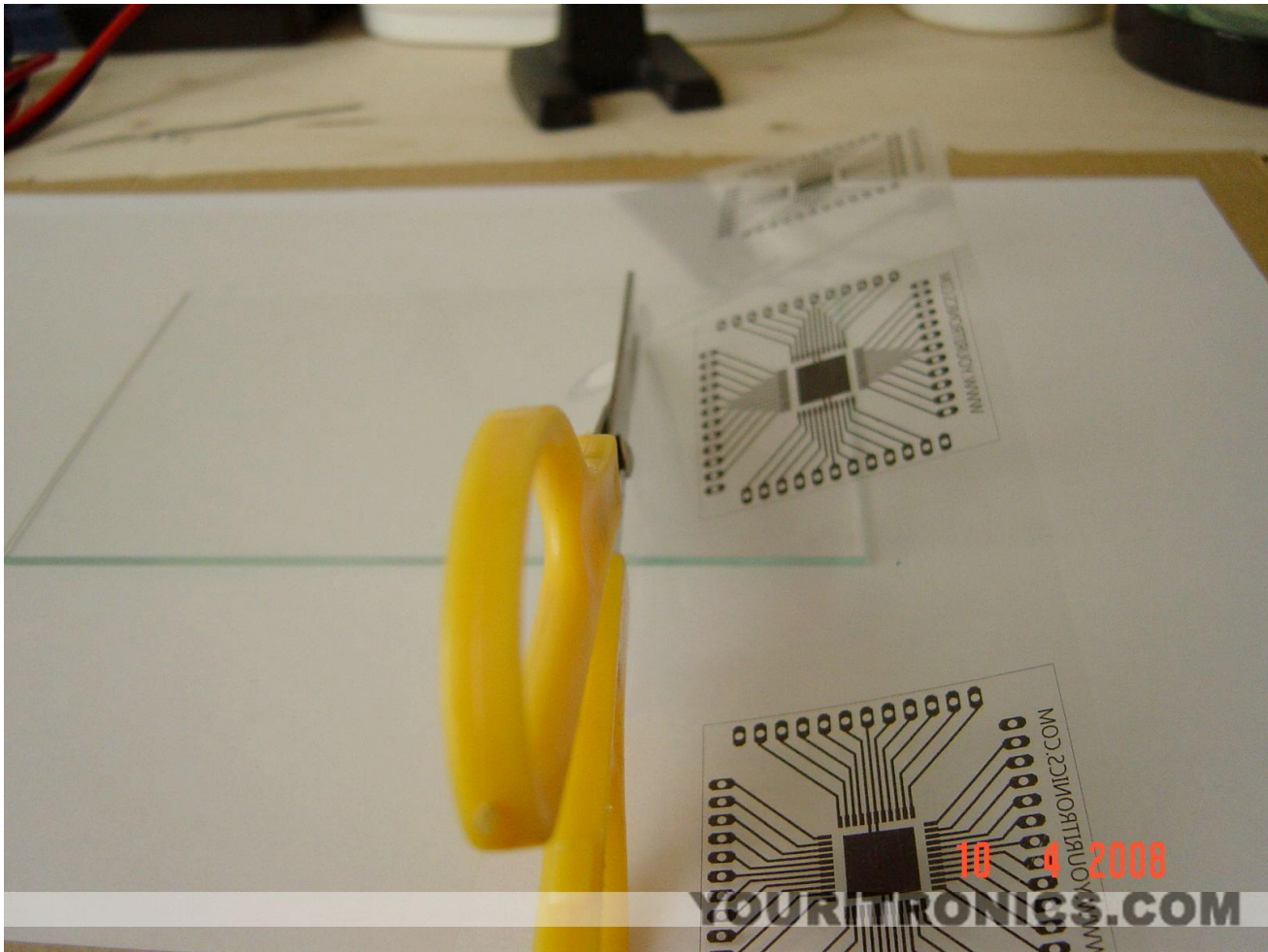
Realizzare PCB – fase 4

Routare i componenti al cad



Realizzare PCB – fase 5

Stampare su carta trasparente – MASTER (circuito specchiato)

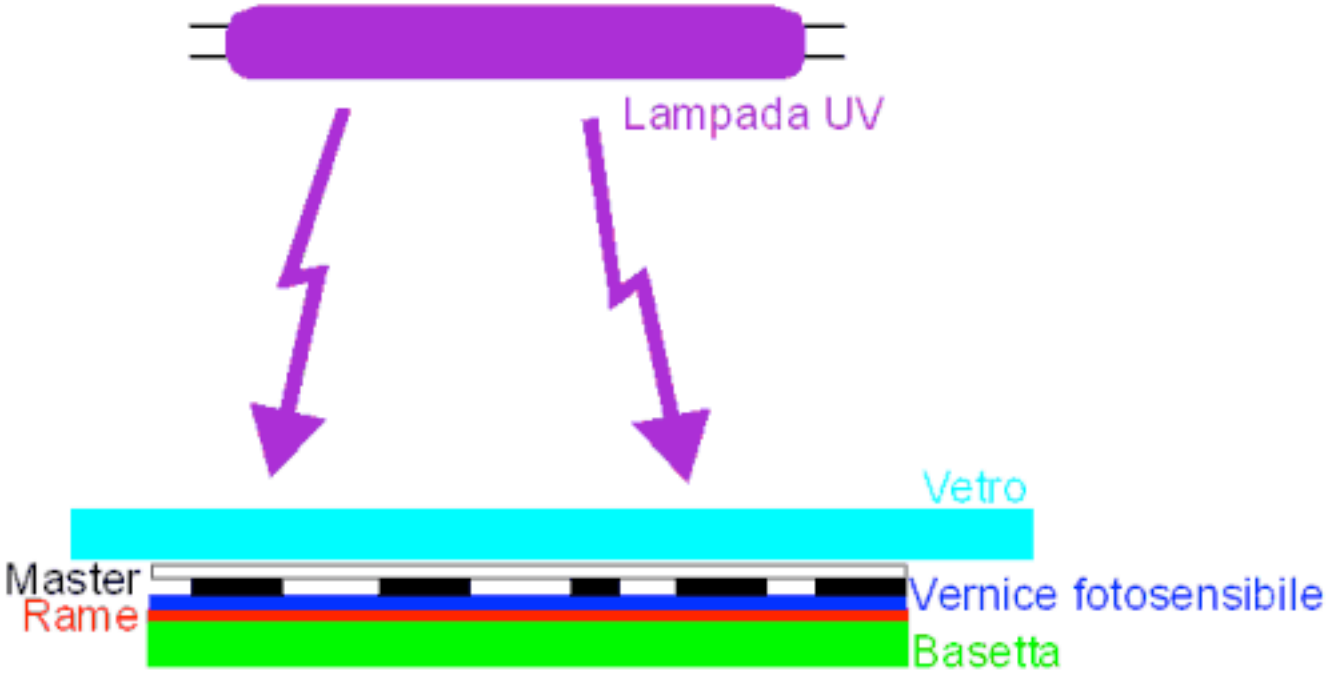


Realizzare PCB – fase 6

Utilizzando delle basette fotosensibili

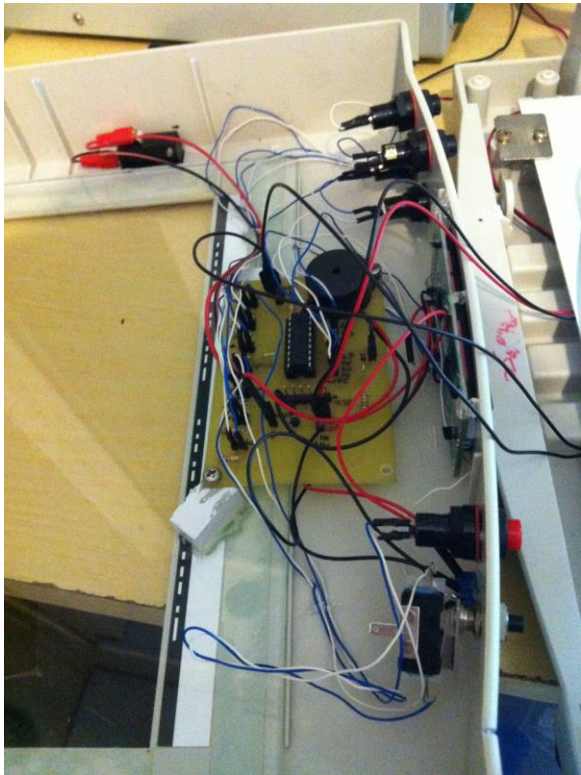
Realizzare PCB – fase 7

Lo sviluppo con il bromografo



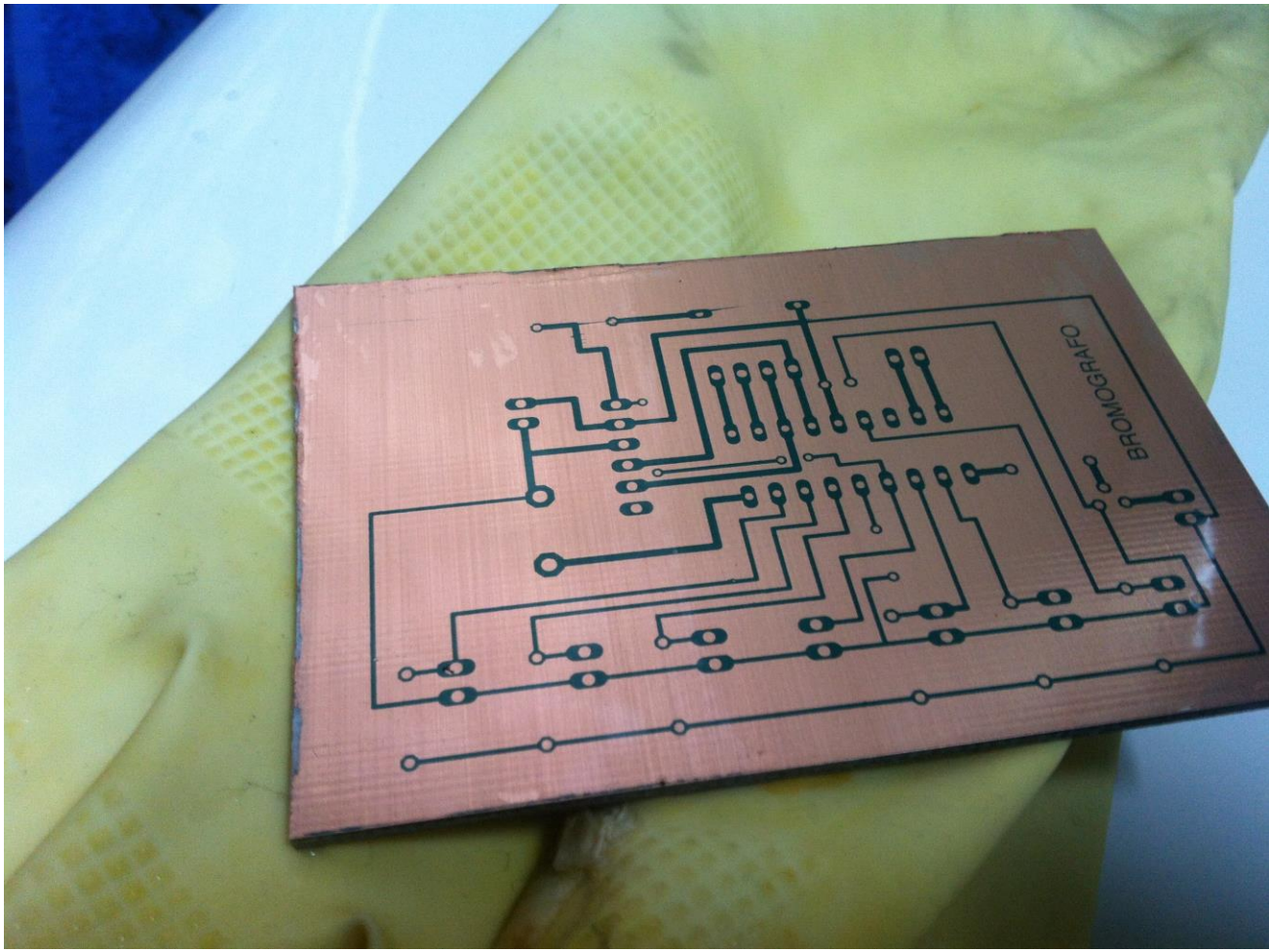
Realizzare PCB – fase 8

Parentesi… il bromografo



Realizzare PCB – fase 9

Dopo il lavaggio con la soda caustica

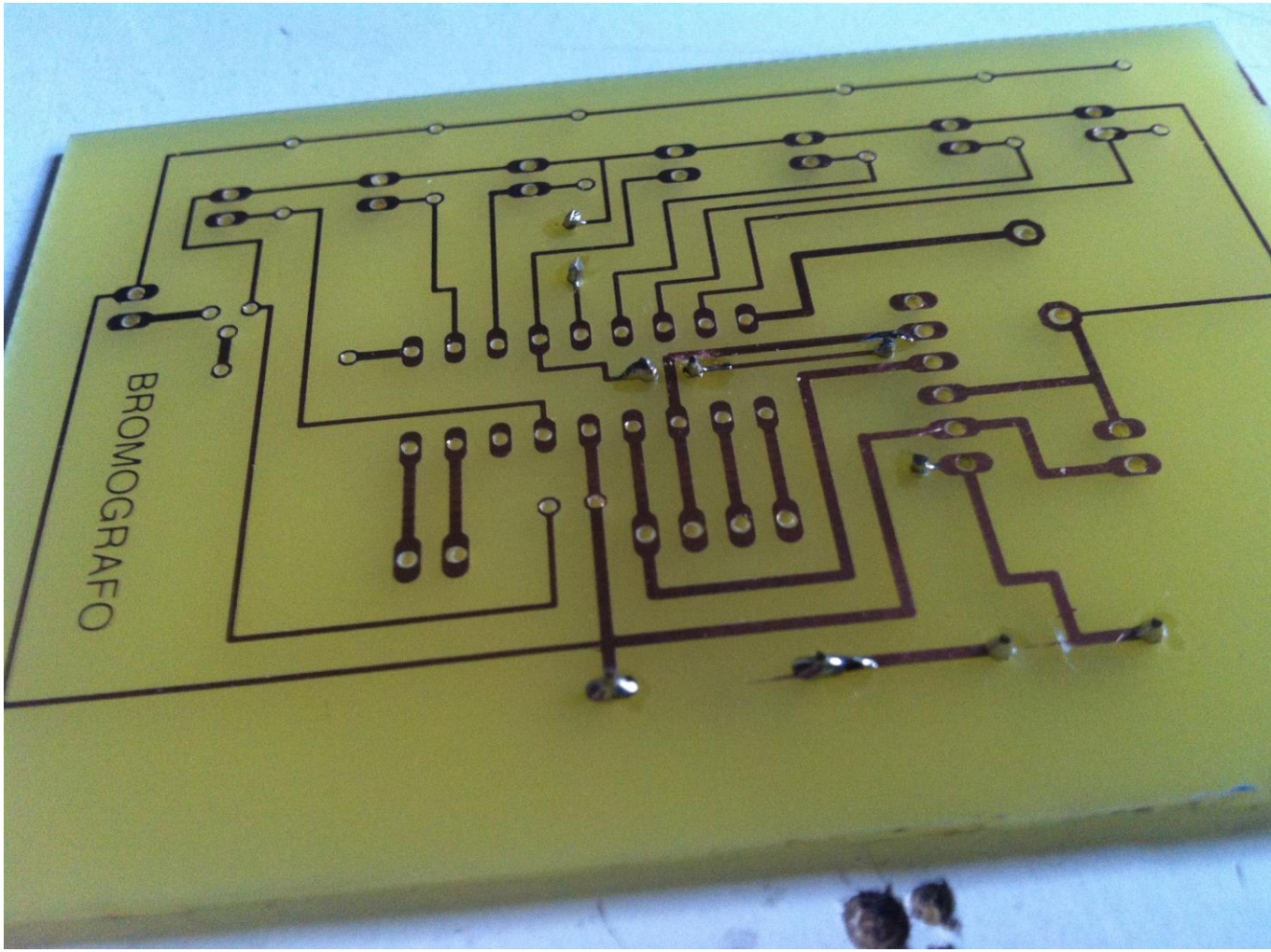


Realizzare PCB – fase 10

Circuito nell’acido

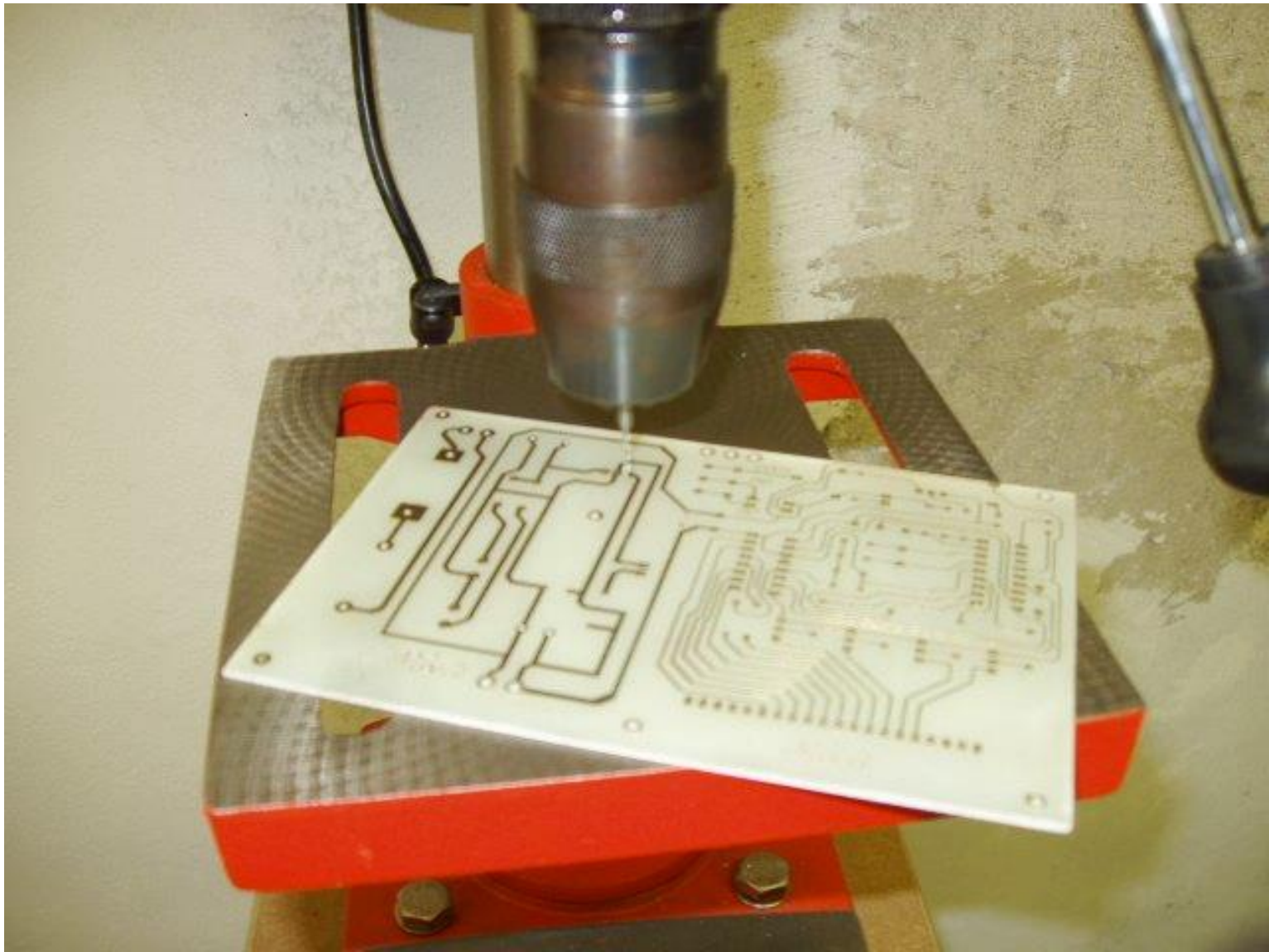
Realizzare PCB – fase 11

Dopo il lavaggio



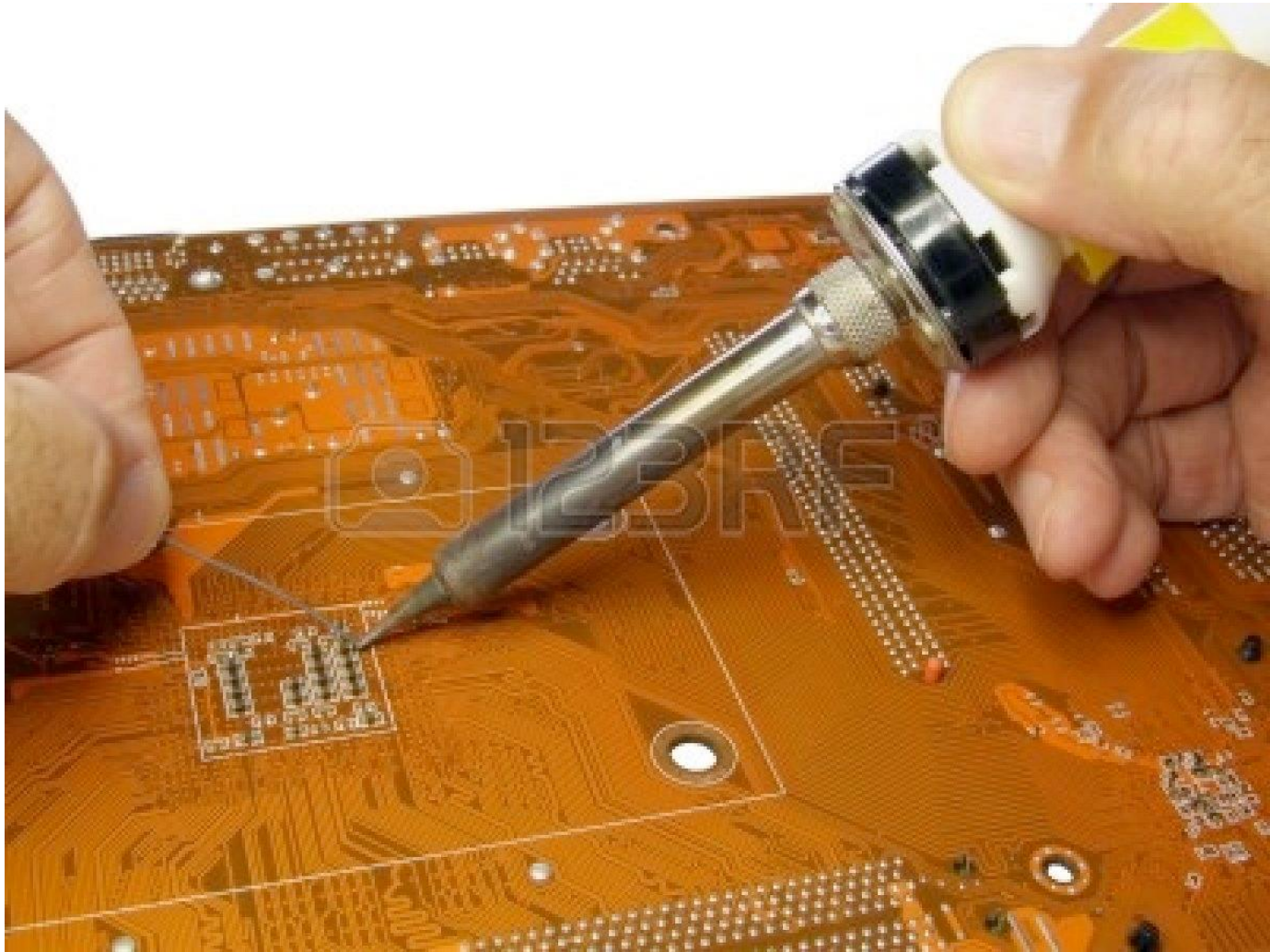
Realizzare PCB – fase 12

foratura

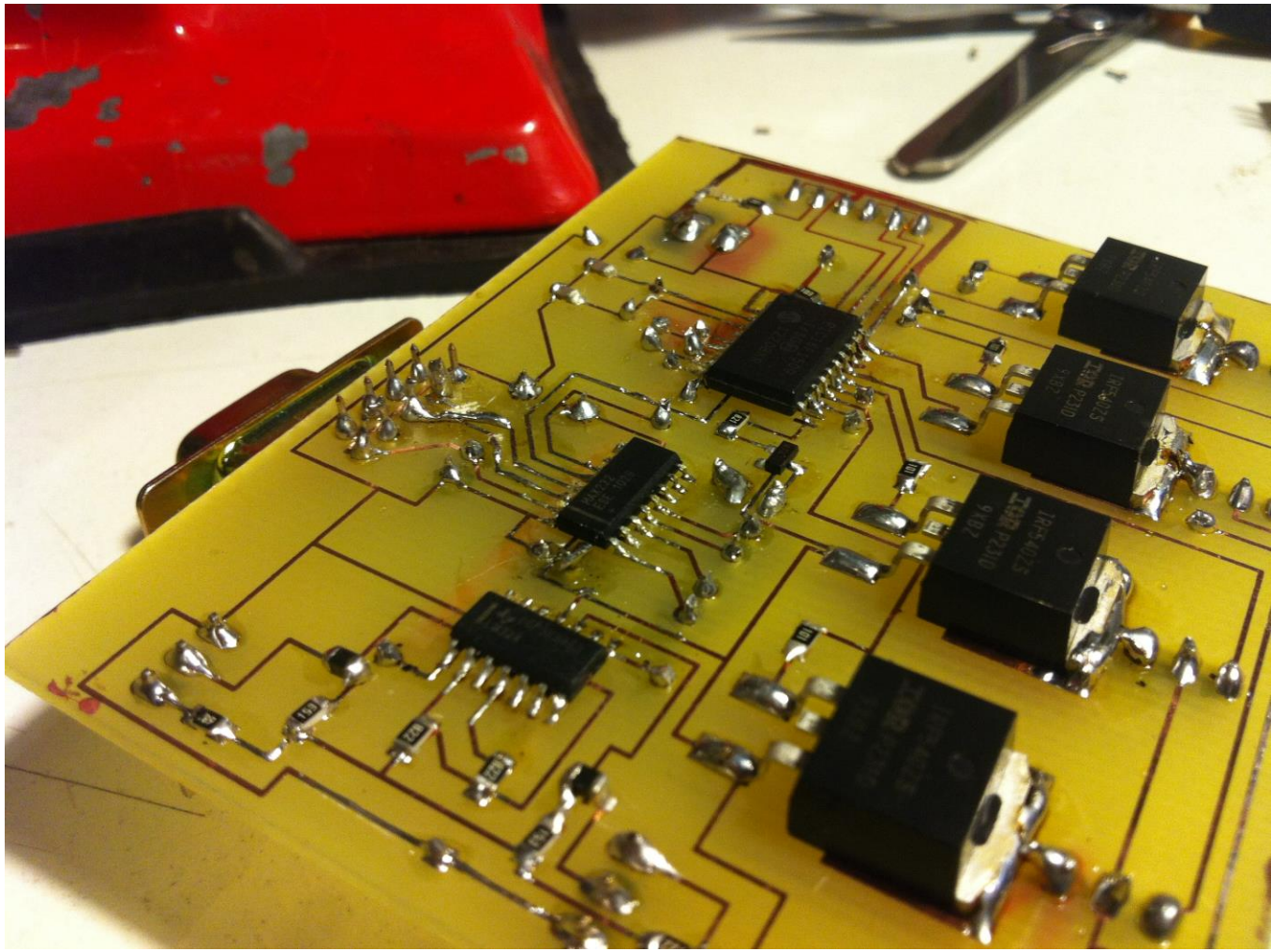


Realizzare PCB – fase 13

saldatura



Alla fine



ACCESSO DIRETTO AI REGISTRI

Accedere direttamente ai registri del microcontrollore ha degli svantaggi

- difficoltà di manutenzione del codice
- perdita di portabilità
- errori

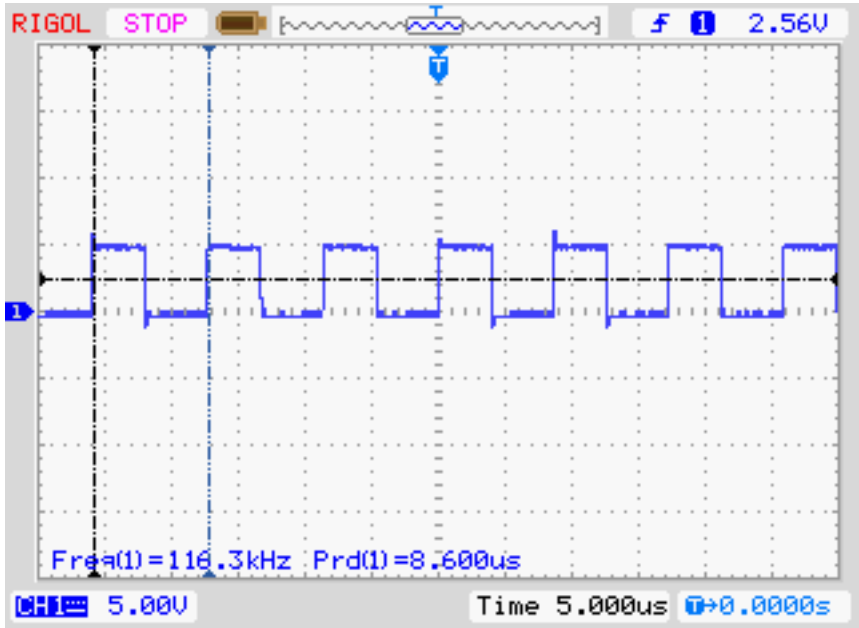
Ma anche dei vantaggi

- facilità/velocità di accesso
- Alcune volte serve di impostare diversi pin nello stesso momento
  PORTB |= B1100;
  (digitalRead() and digitalWrite() sono composte a molte righe di codice)

ESEMPIO: DIGITALWRITE

```
void digitalWrite(uint8_t pin, uint8_t val) {
  uint8_t timer, bit, port, oldSREG;
  volatile uint8_t *out;

  //timer = digitalPinToTimer(pin);
  timer = pgm_read_byte(digital_pin_to_timer_PGM + pin );
  //bit = digitalPinToBitMask(pin);
  bit = pgm_read_byte( digital_pin_to_bit_mask_PGM + pin );
  //port = digitalPinToPort(pin);
  port = pgm_read_byte( digital_pin_to_port_PGM + pin );

  if (port == NOT_A_PIN)
    return;

  //If the pin that support PWM output, we need to turn it off
  //before doing a digital write.
  if (timer != NOT_ON_TIMER)
    turnOffPWM(timer);

  //out = portOutputRegister(port);
  out =(volatile uint8_t *)(pgm_read_word( port_to_output_PGM + pin ));

  oldSREG = SREG;
  cli();

  if (val == LOW)
    *out &= bit;  //clear bit
  else
    *out |= bit;  //set bit

  SREG = oldSREG;
}
```

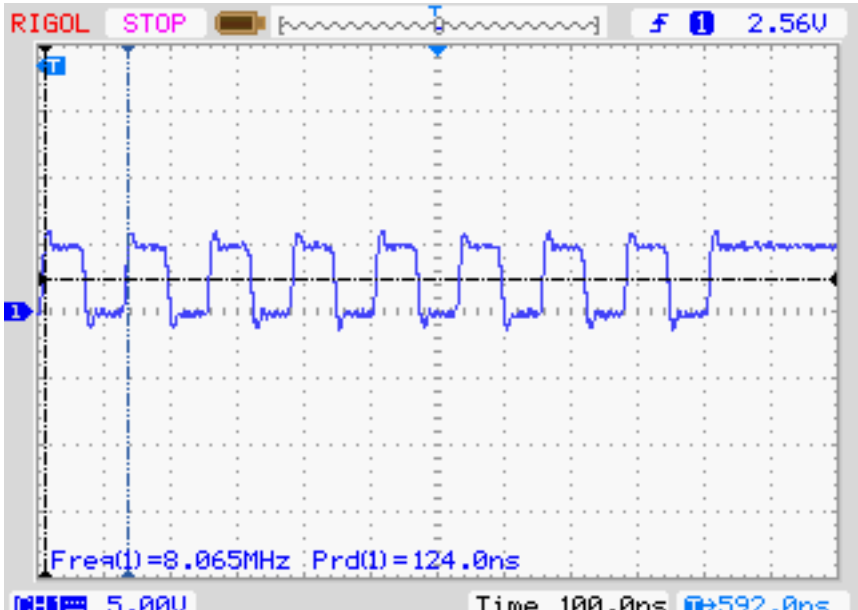
Risultato



...e port manipulation

```
PORTB =0;
PORTB =B100000;
```

Risultato



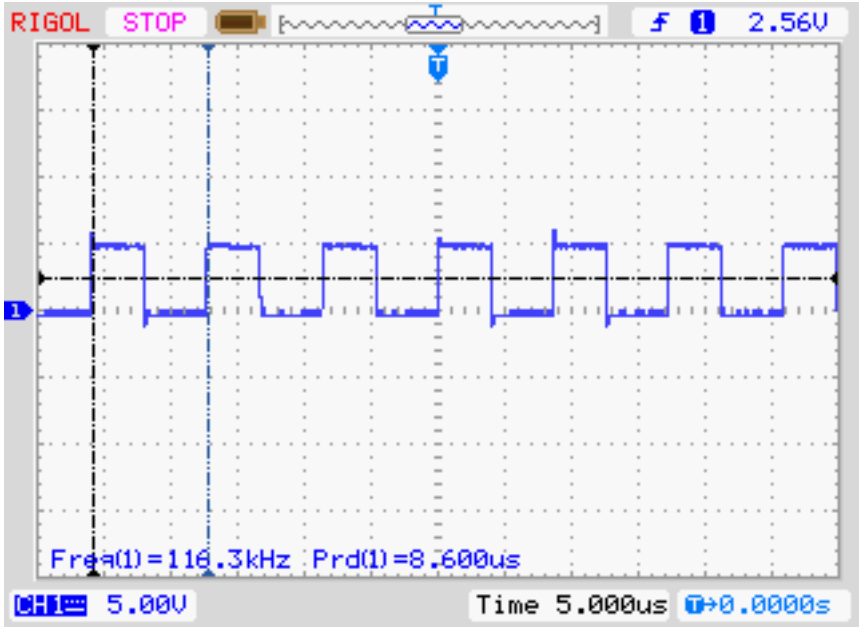
Importante: inizializzazione delle variabili

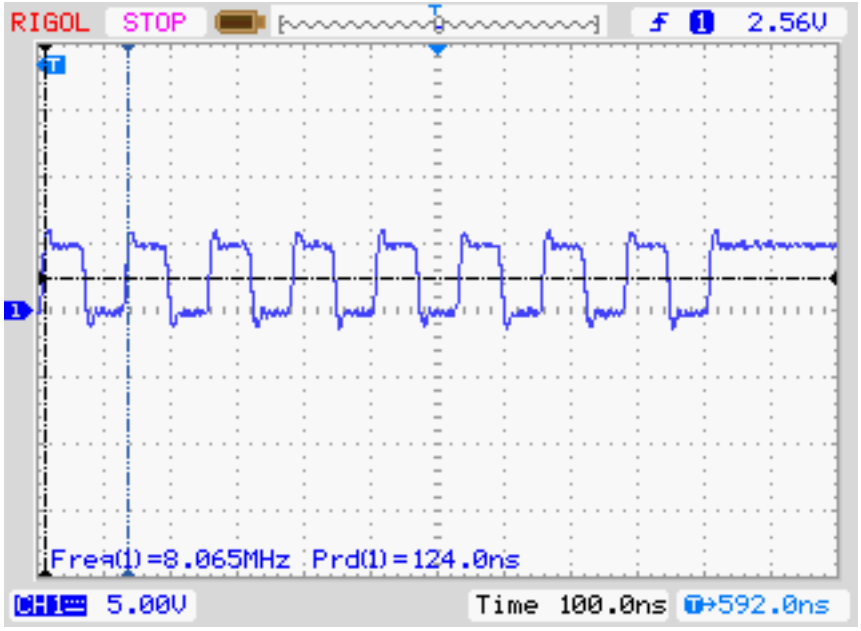
A = B00000101 -> B binario

A = 0x05 -> 0x esadecimale

A = 5 -> decimale

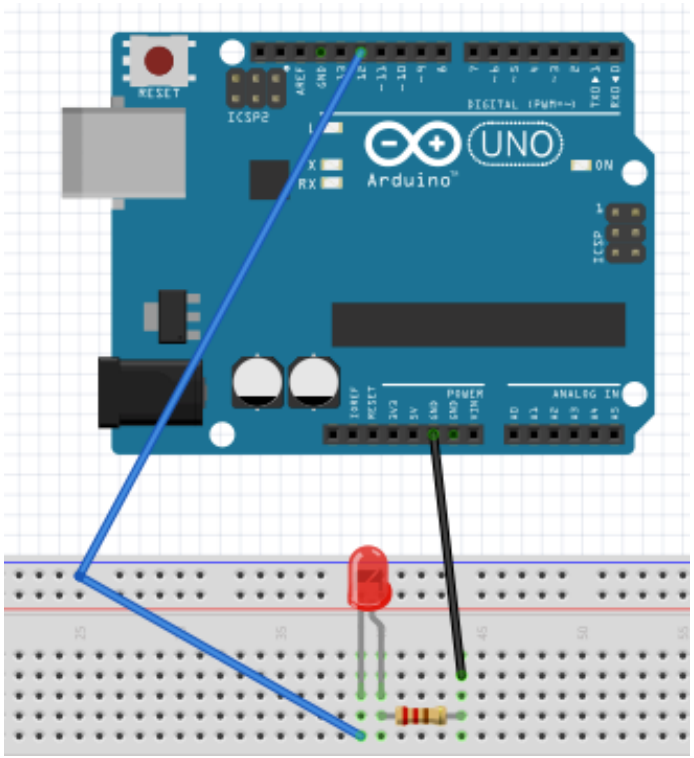
Confrontando...



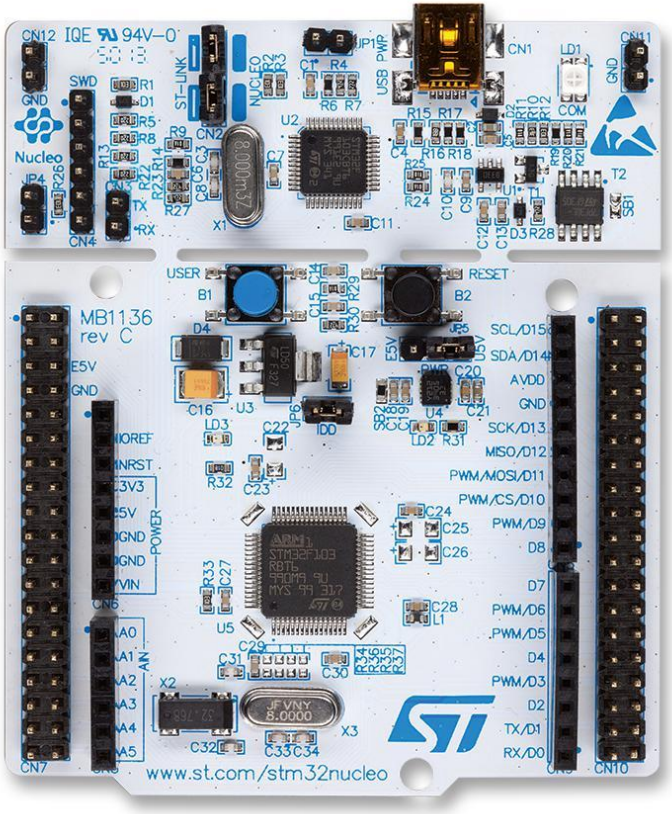


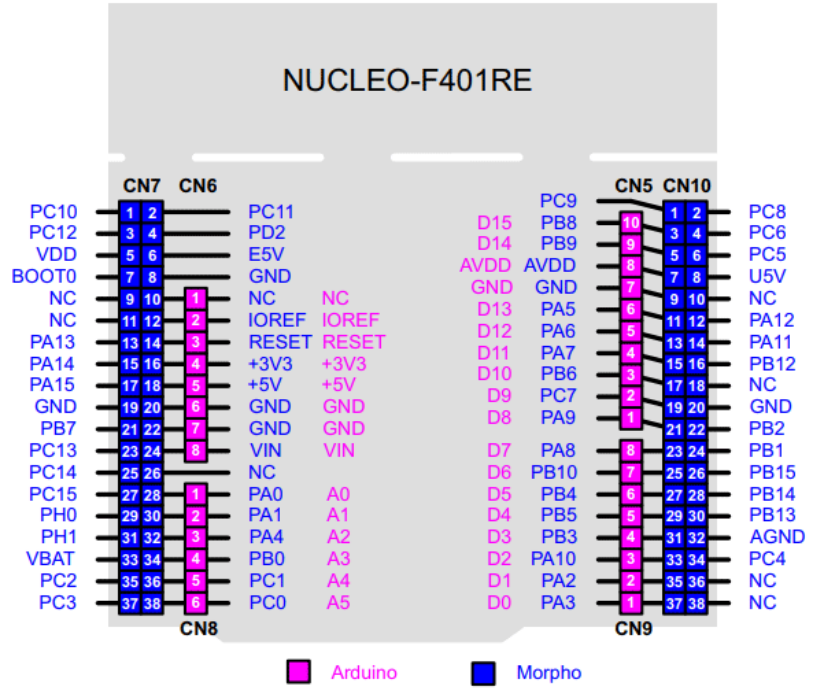
ACCENDERE UN LED CON PORT MANIPULATION

```
void setup(){
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop(){
  //digitalWrite(7, HIGH);
  //digitalWrite(6, HIGH);
  //digitalWrite(5, HIGH);
  PORTD = PORTD | B11100000;
  delay(1000);

  //digitalWrite(7, LOW);
  //digitalWrite(6, LOW);
  //digitalWrite(5, LOW);
  PORTD = PORTD & B00011111;
  delay(1000);
}
```

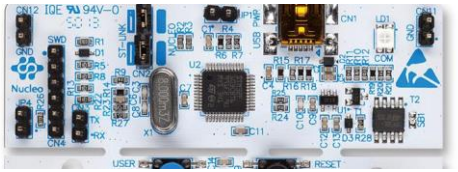


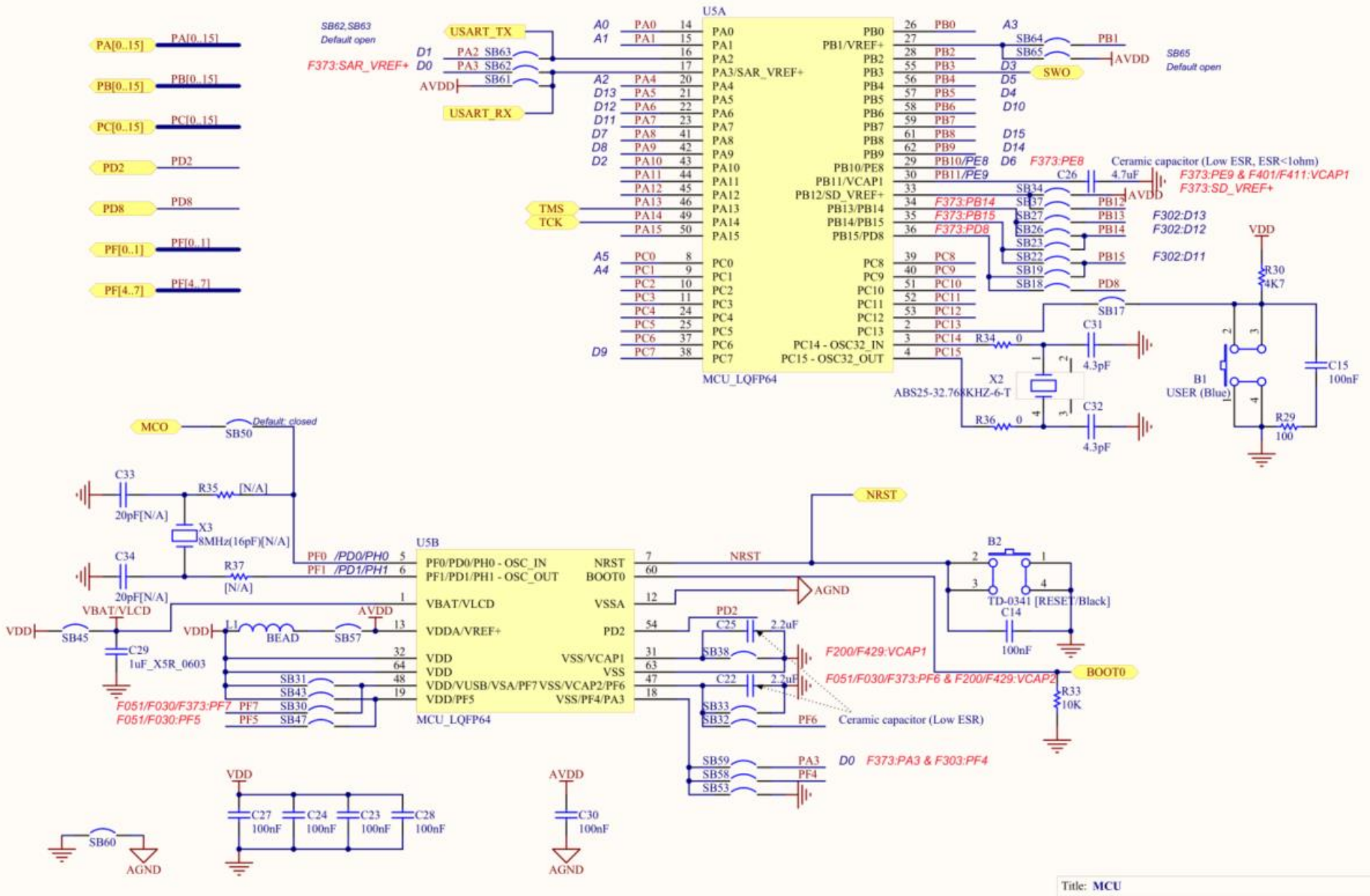
NUCLEO F401RE - HW



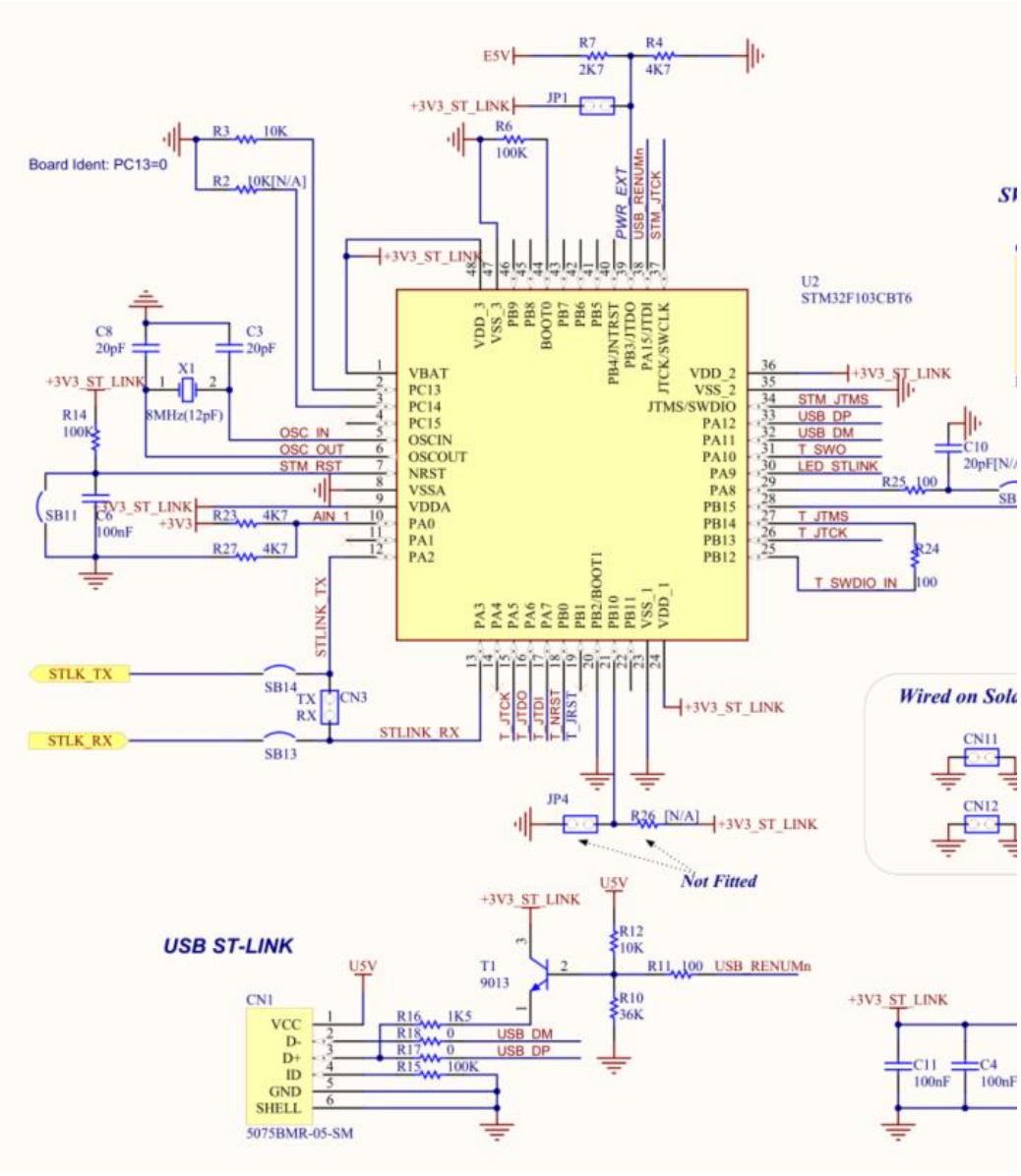


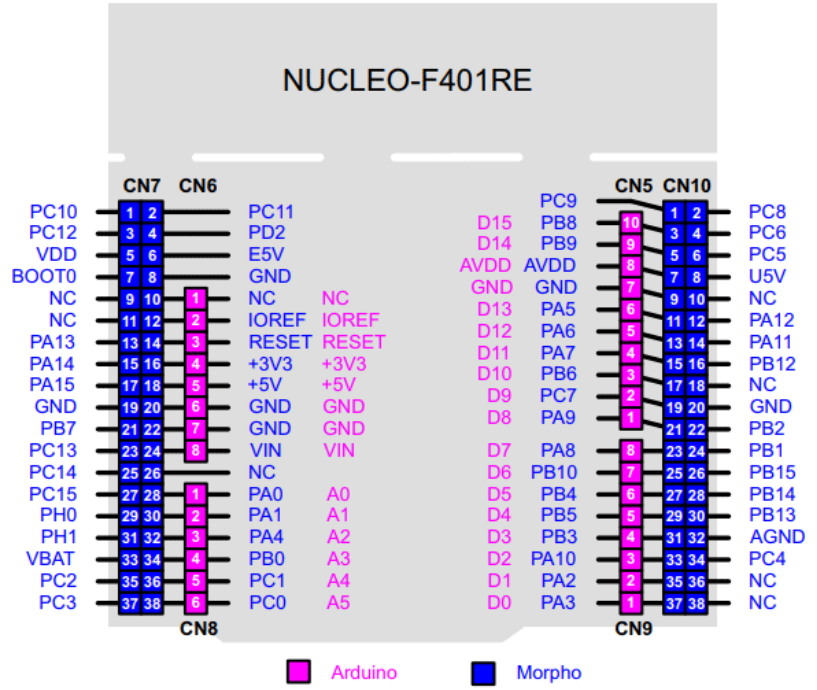
NUCLEO F401RE - SCHEMA



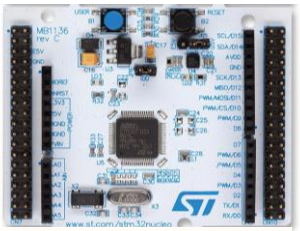


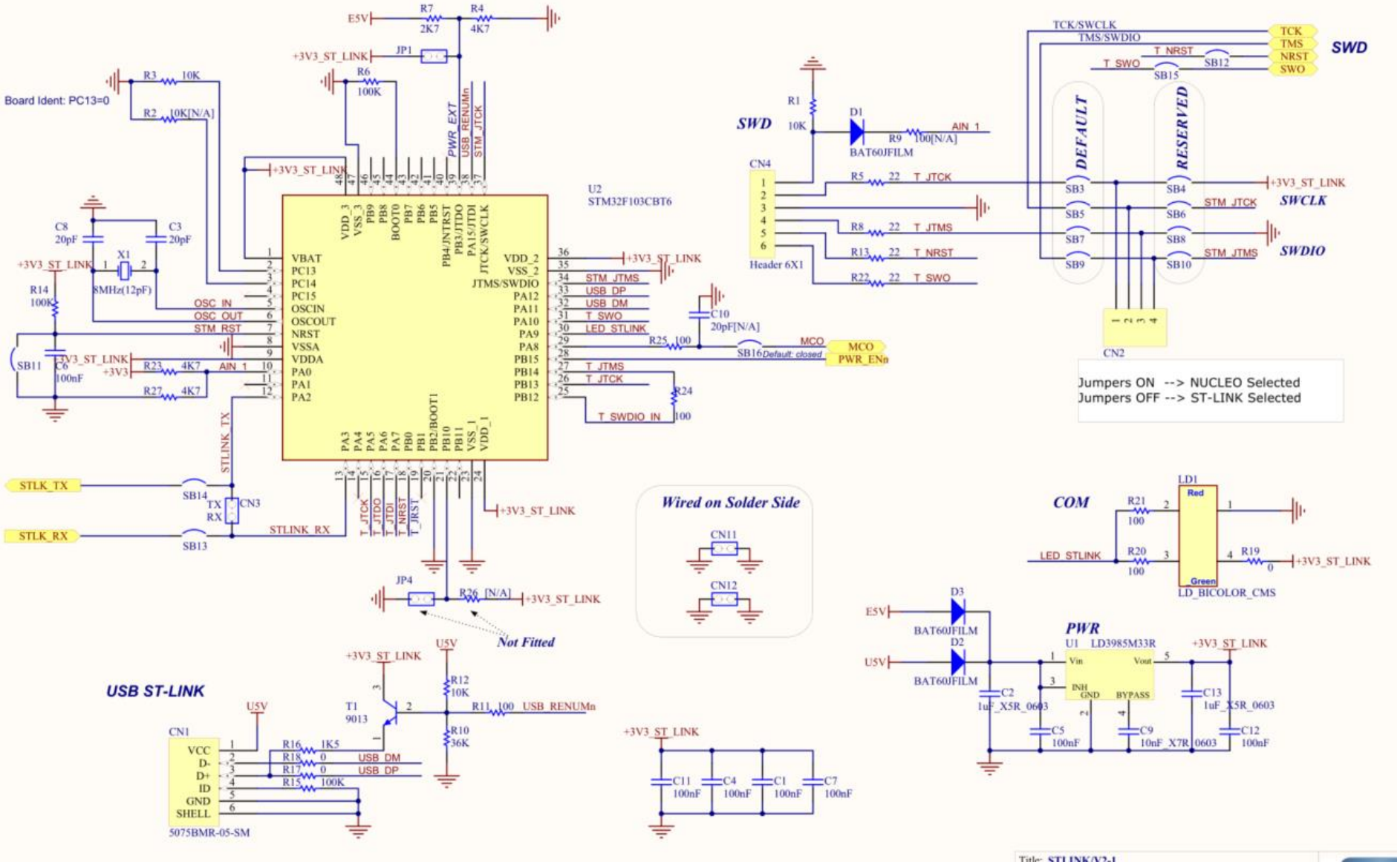
NUCLEO F401RE - SCHEMA





NUCLEO F401RE - SCHEMA





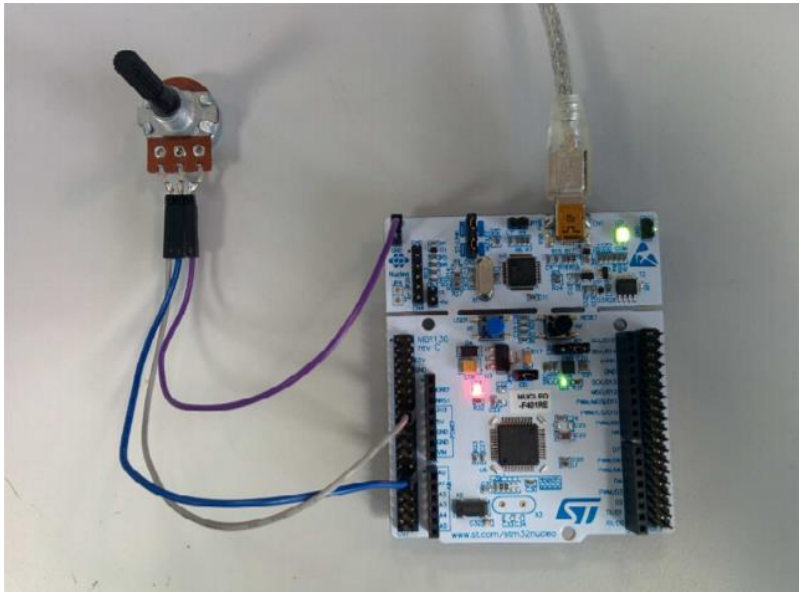
NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

PWM => per pilotare il LED
Timer 2 Channel 1 - PWM mode • 100Hz •

ADC => per variare la luminosità del LED variando proporzionalmente al valore analogico acquisito il Duty Cycle del PWM

- ADC1 IN0 – Single and regular conversion

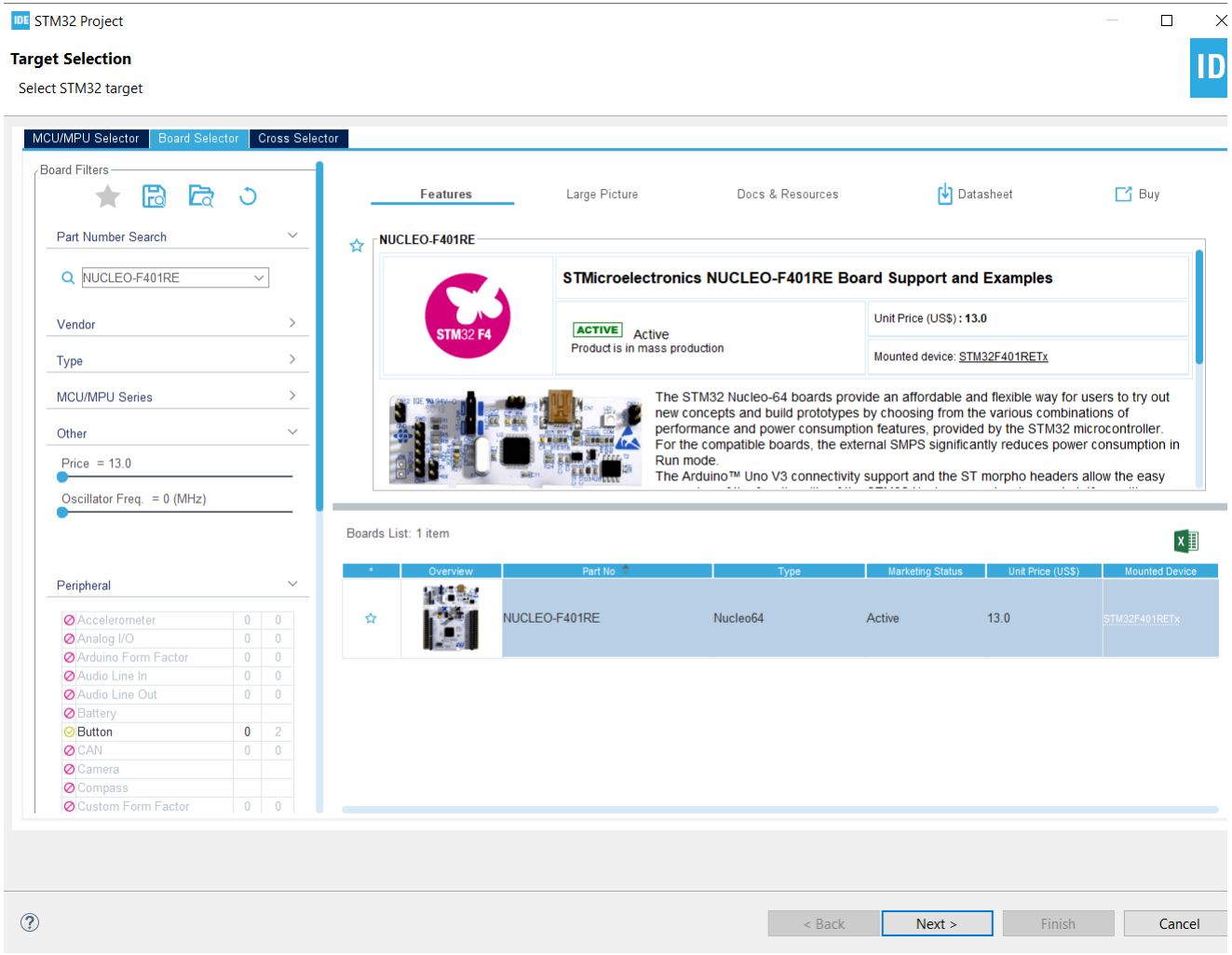


AVVERTIMENTO:
Sarà PARECCHIO più complicato rispetto a scrivere «analogWrite»

NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

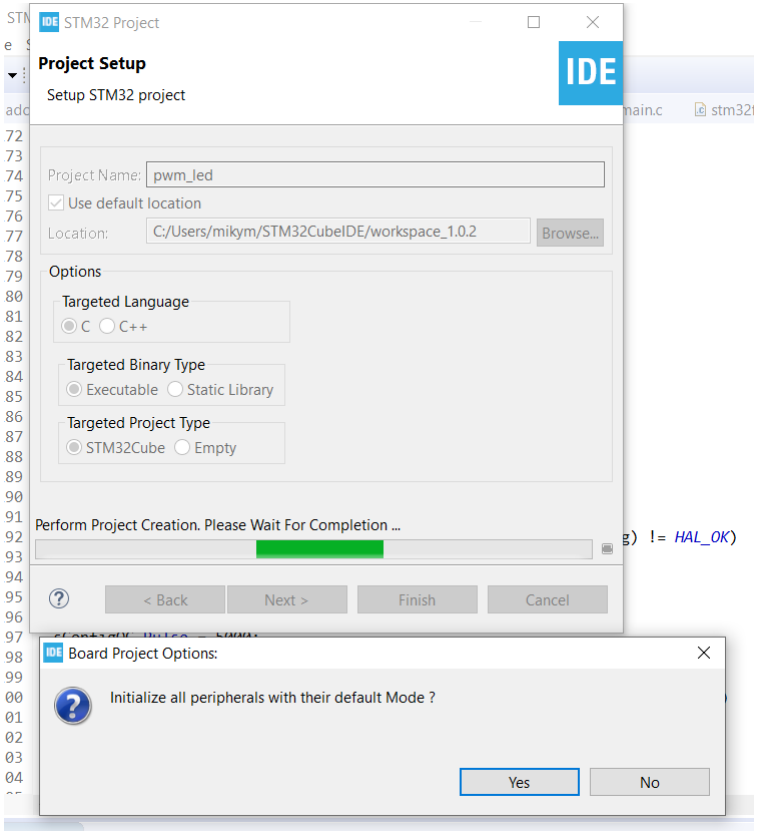
File -> new -> project -> STM32project -> board selector -> nucleo F401RE



NUCLEO F401RE – ESEMPIO

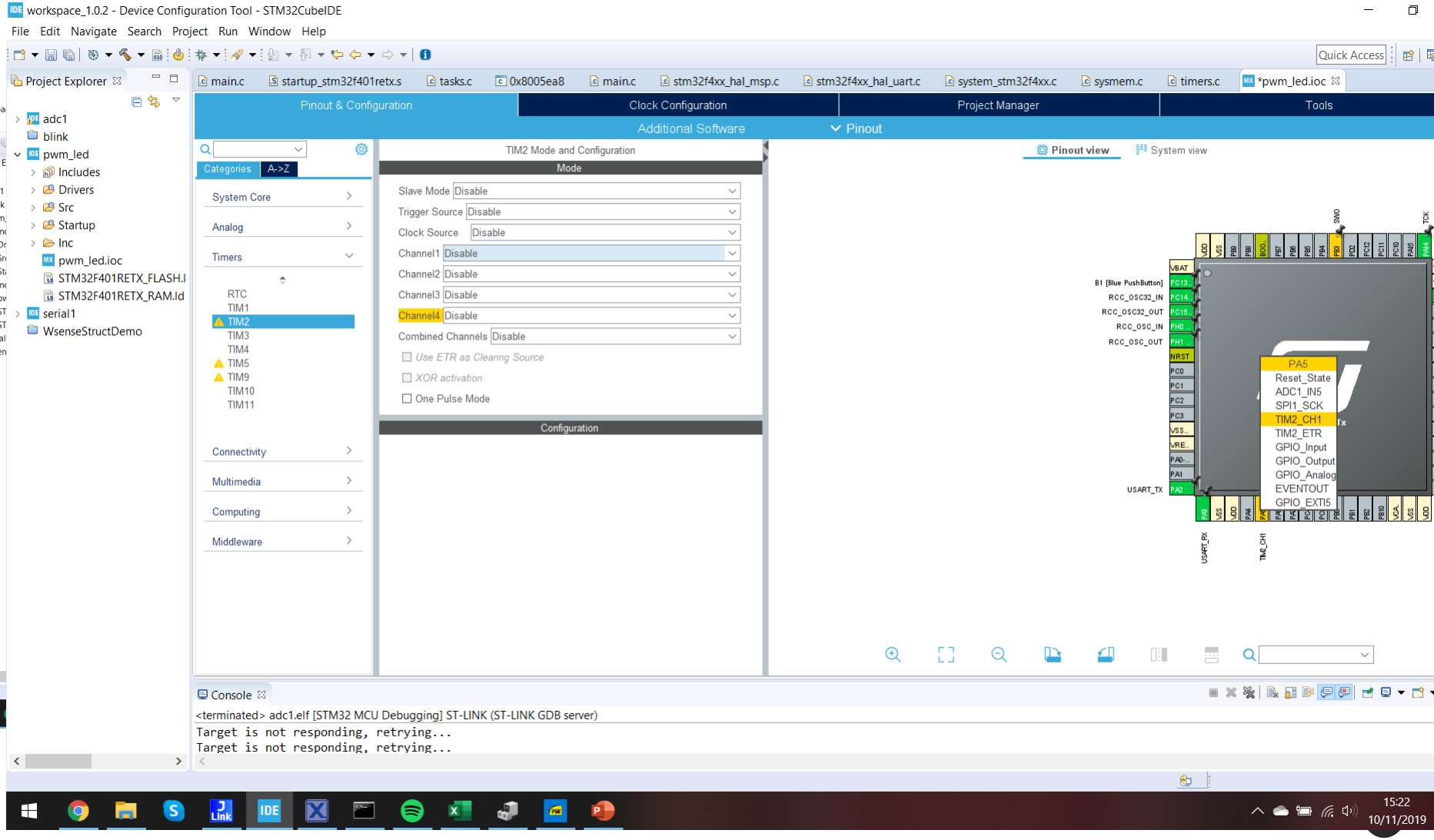
USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

Assegnare nome – e poi yes



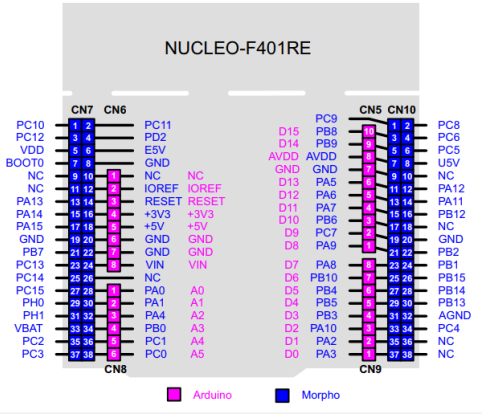
NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM



NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN F



Schermata pin

ADC

Timer

onfiguration Tool - STM32CubeIDE

Project Run Window Help

main.c | startup_stm32f401retx.s | tasks.c | 0x8005ea8 | main.c | stm32f4xx_hal_msp.c | stm32f4xx_hal_uart.c | system_stm32f4xx.c | sysmem.c | timers.c | pwm_led.ioc

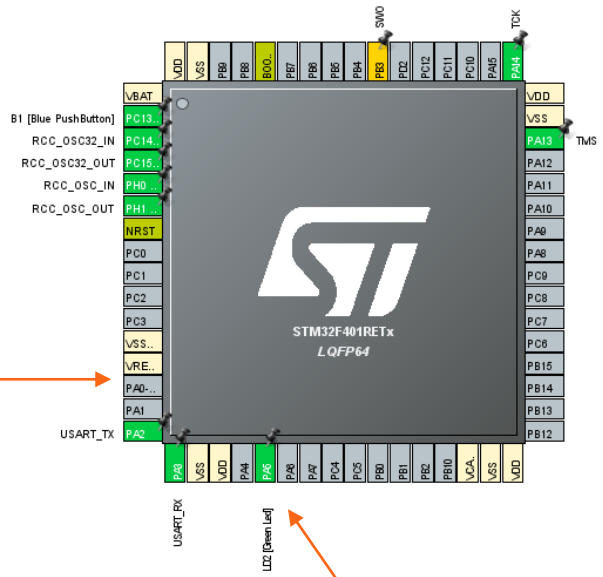
Pinout & Configuration | Clock Configuration | Project Manager | Tools

Additional Software | Pinout

Pinout view | System view

Categories | A->Z

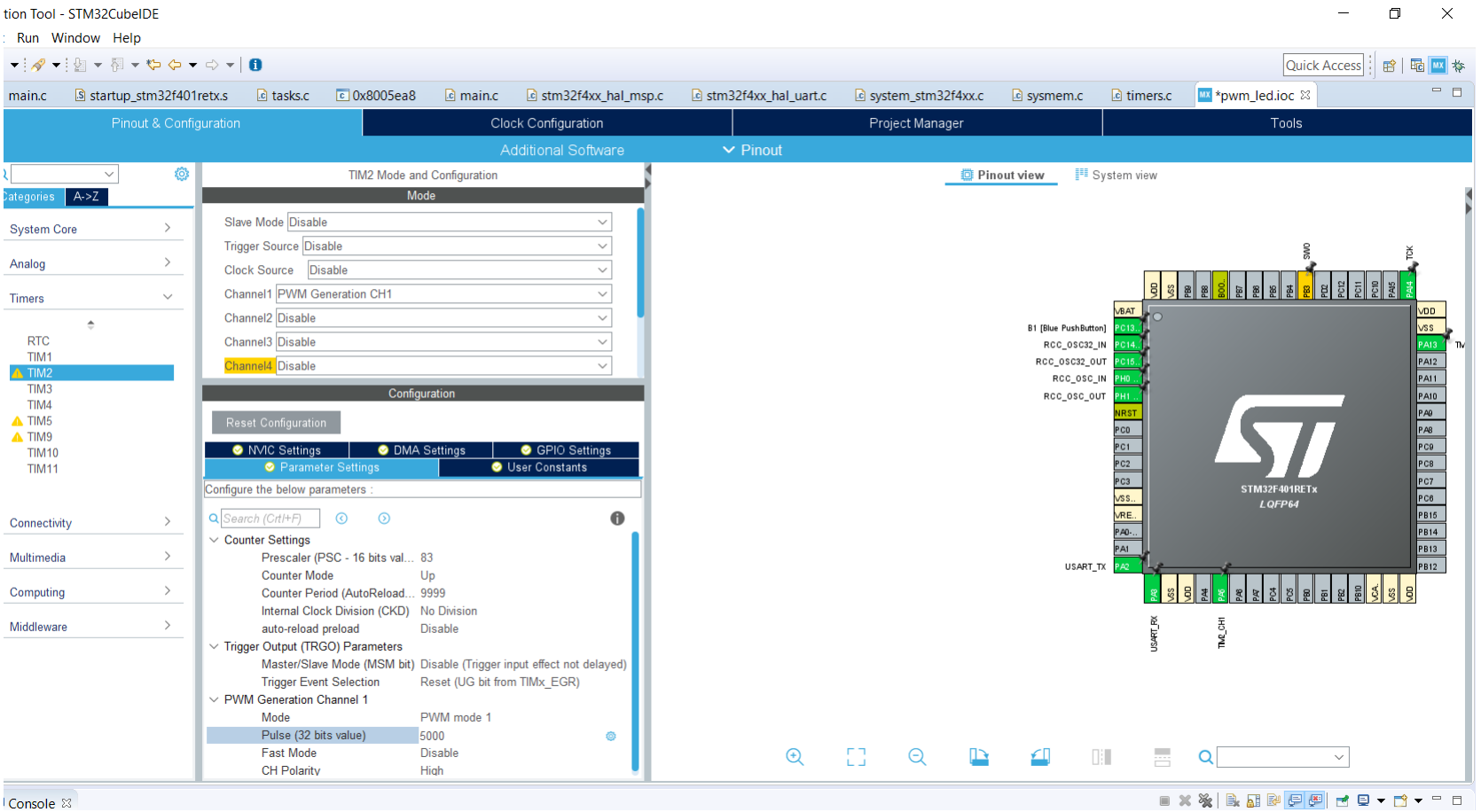
- System Core
- Analog
- Timers
- Connectivity
- Multimedia
- Computing
- Middleware



NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

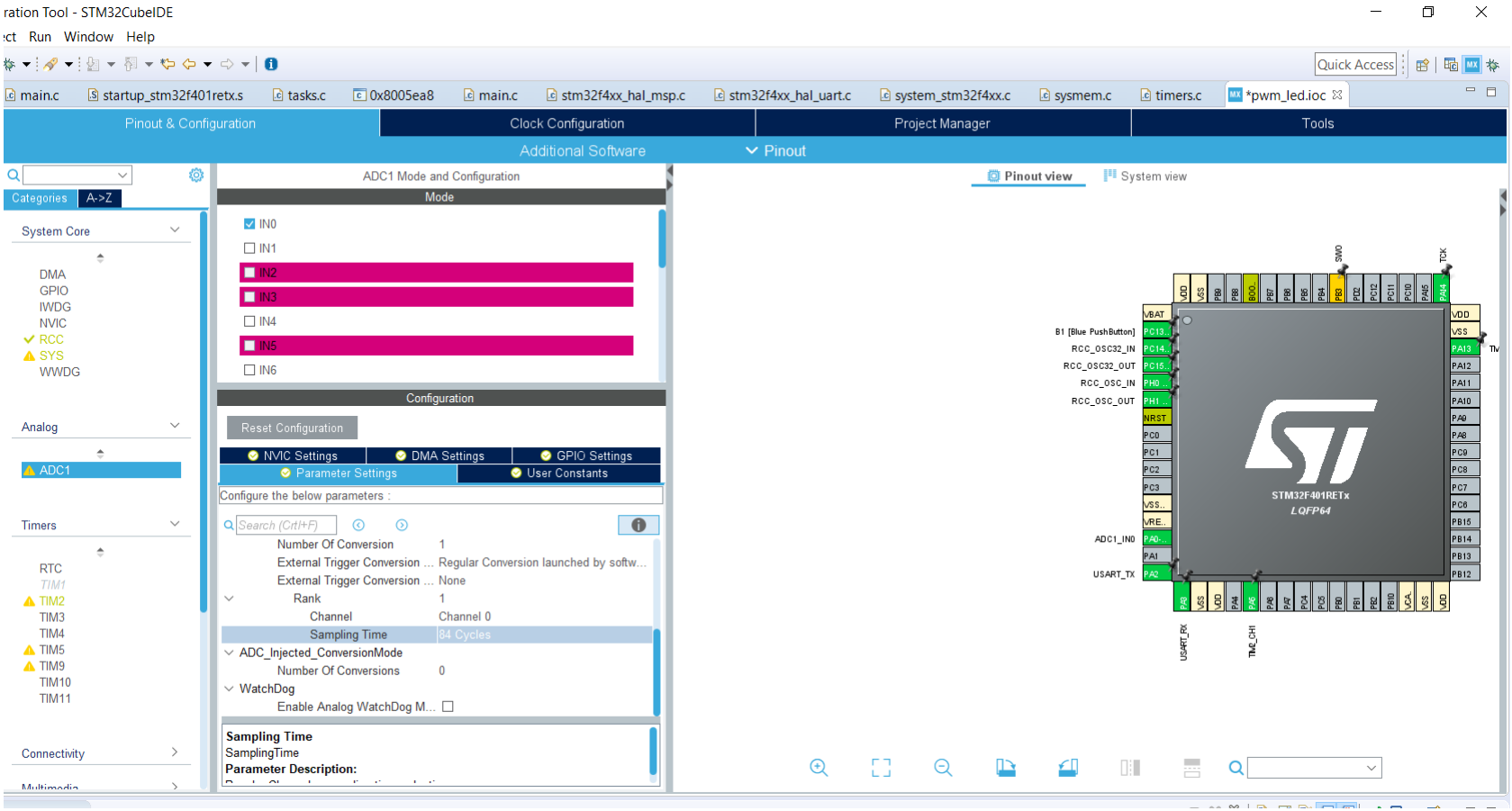
Proprietà del timer: prescaler 83, period 9999 (100hz), pulse 5000 (50%)



NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

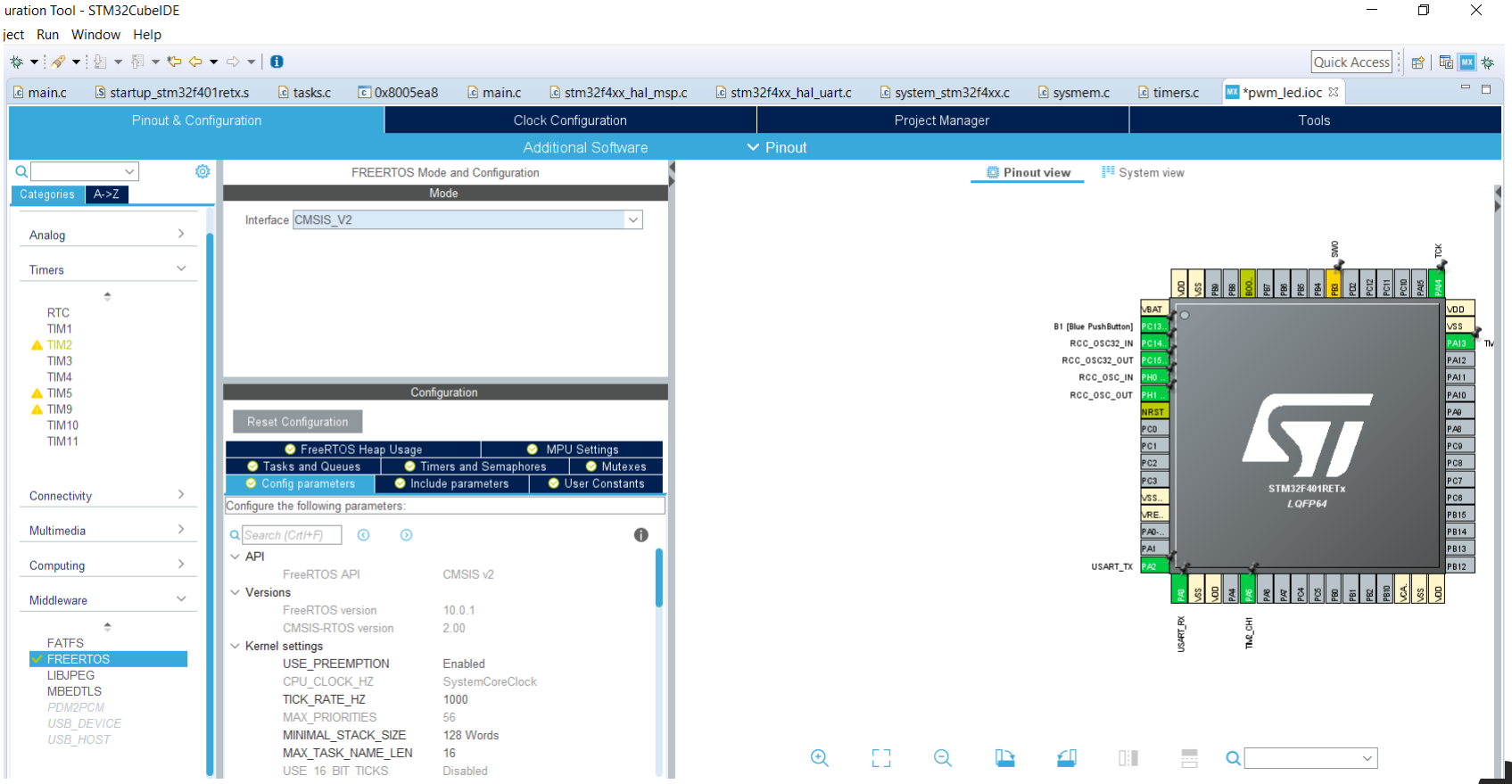
Adc: canale 0 e 84 cicli a conversione



NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

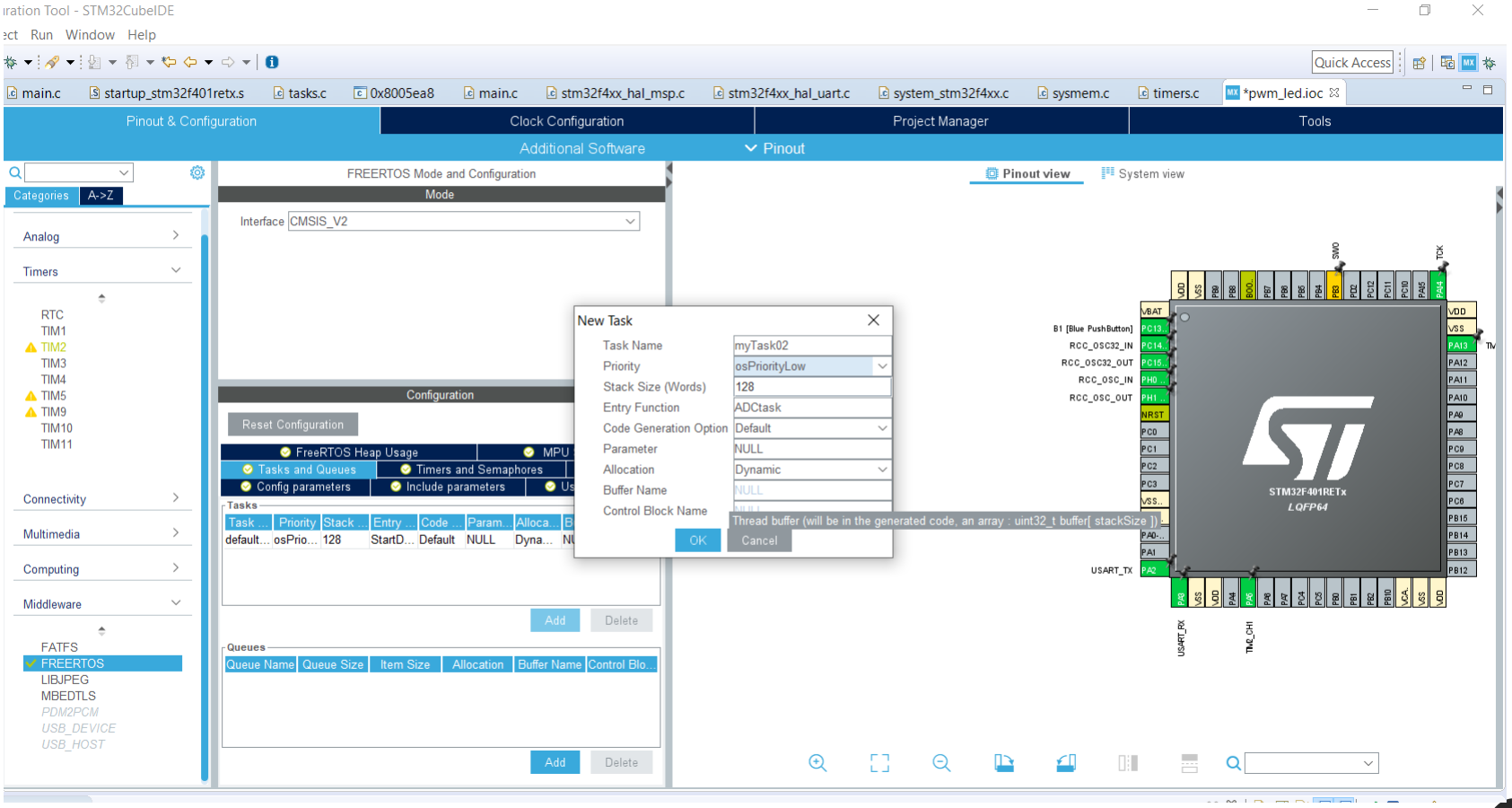
Middleware - Freertos



NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

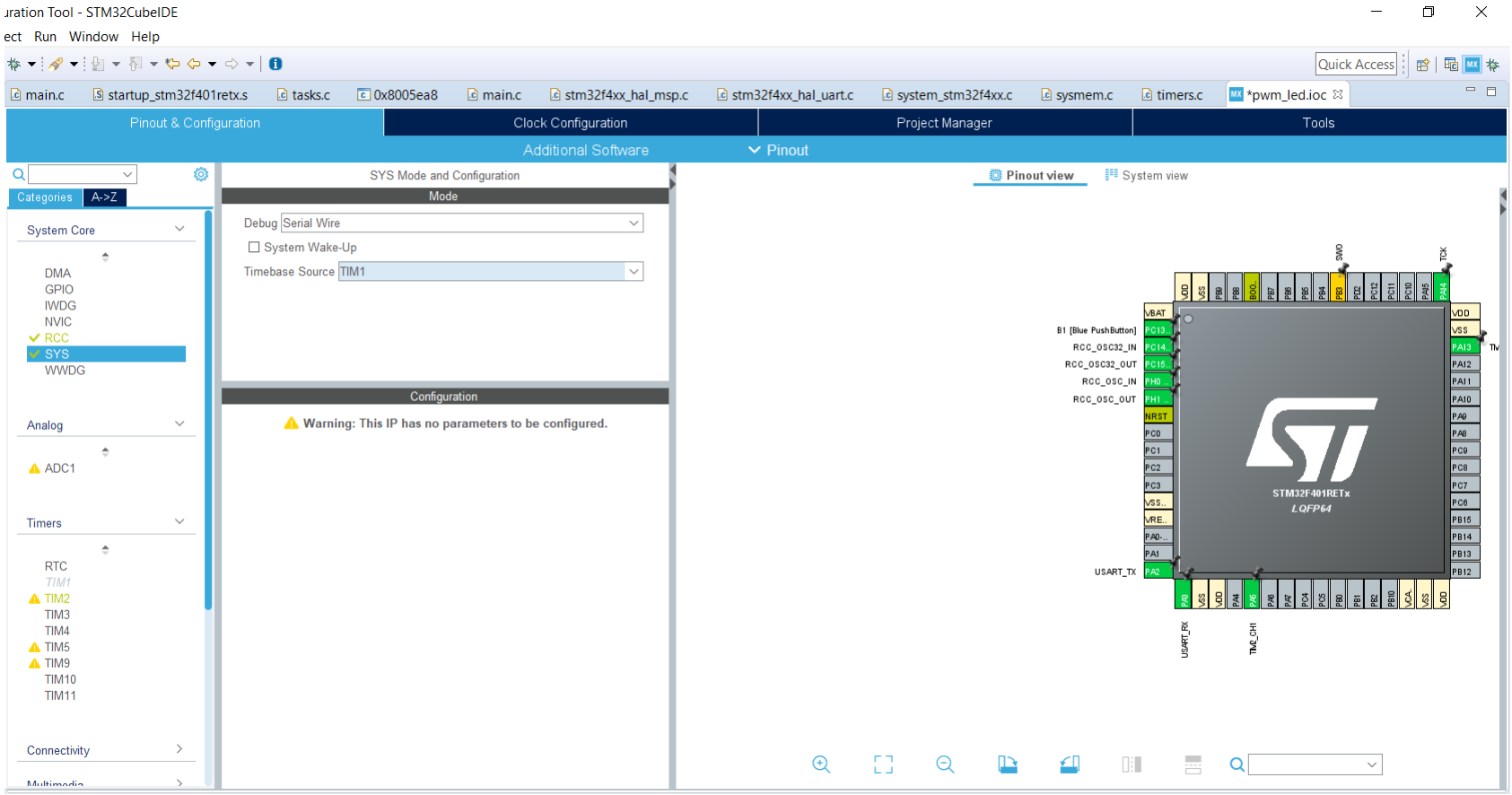
Freertos – creazione task



NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

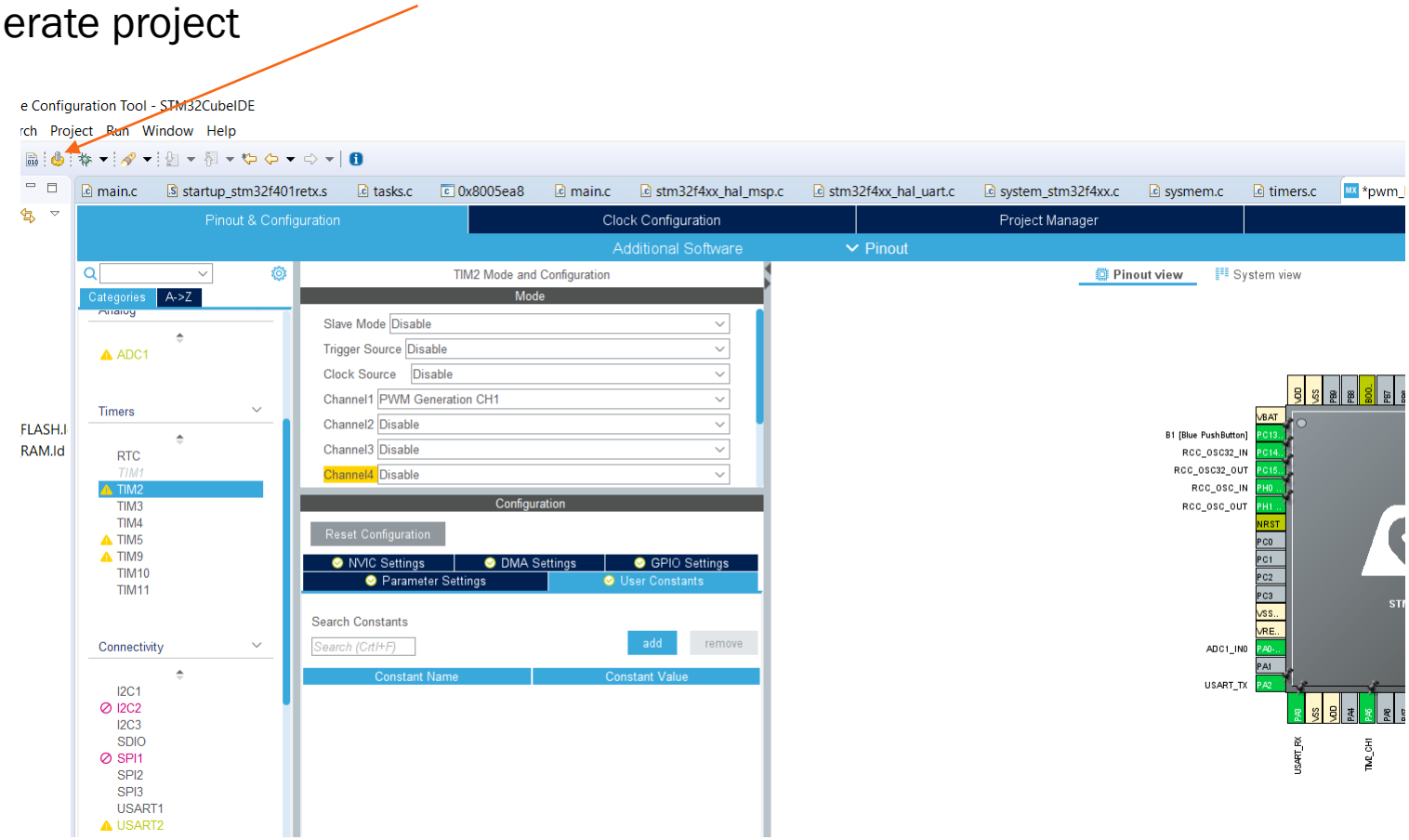
Freertos – systick



NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

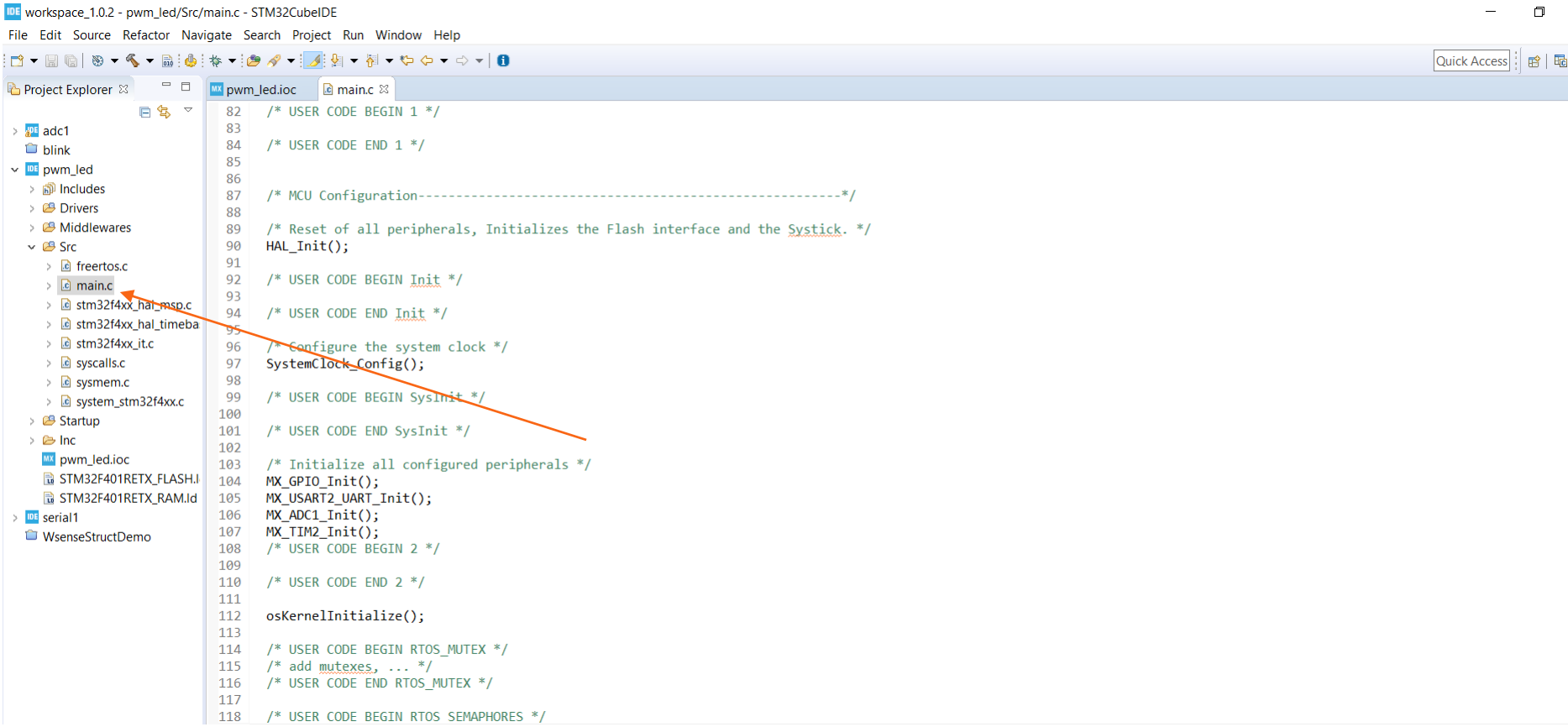
Generate project



NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

Aprire main.c



NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

Inizializzare i PWM (questo è un TIMER)

ain.c - STM32CubeIDE

avigate Search Project Run Window Help

Quick Access

pwm_led.ioc | main.c | main.c

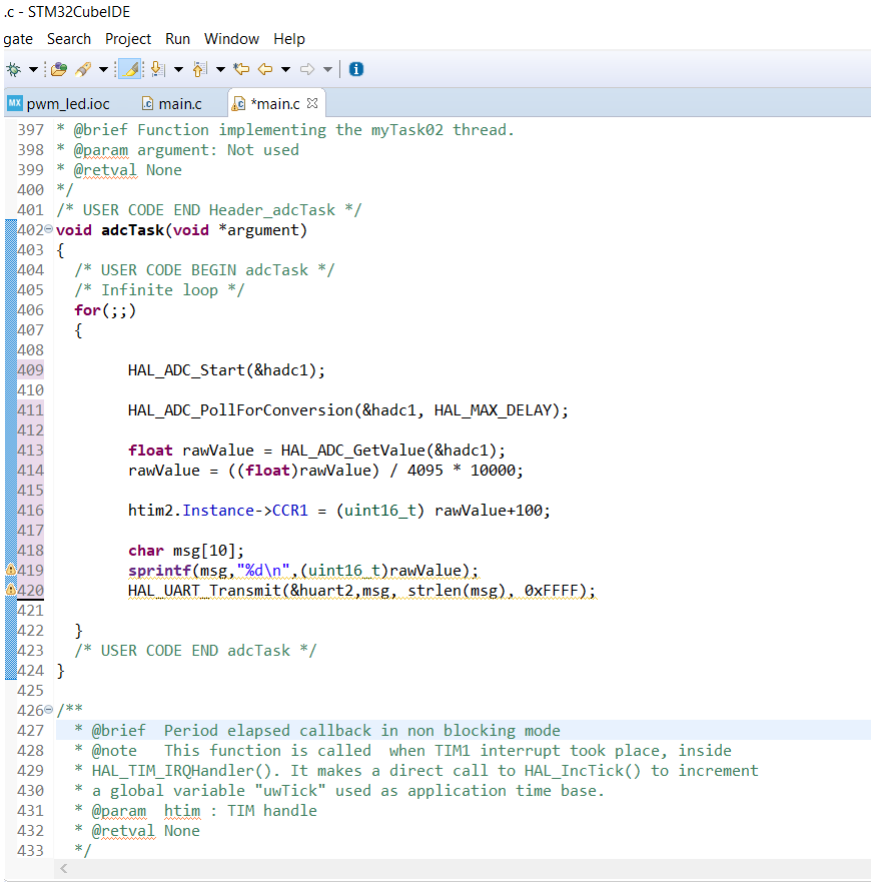
```
136   };
137   defaultTaskHandle = osThreadNew(StartDefaultTask, NULL, &defaultTask_attributes);
138
139   /* definition and creation of myTask02 */
140   const osThreadAttr_t myTask02_attributes = {
141     .name = "myTask02",
142     .priority = (osPriority_t) osPriorityLow,
143     .stack_size = 128
144   };
145   myTask02Handle = osThreadNew(adcTask, NULL, &myTask02_attributes);
146
147   /* USER CODE BEGIN RTOS_THREADS */
148   /* add threads, ... */
149   /* USER CODE END RTOS_THREADS */
150
151   /* Start scheduler */
152   HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
153  osKernelStart();
154
155   /* We should never get here as control is now taken by the scheduler */
156
157   /* Infinite loop */
158
159   /* USER CODE BEGIN WHILE */
160   while (1)
161   {
162     /* USER CODE END WHILE */
163
164     /* USER CODE BEGIN 3 */
165   }
166   /* USER CODE END 3 */
167 }
168
169 /**
170   * @brief System Clock Configuration
171   * @retval None
```

NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

Inserire il task

Inizializza l'ADC
Aspetta la conversione del valore
Utilizza il valore per il PWM
Scrive le informazioni sulla seriale

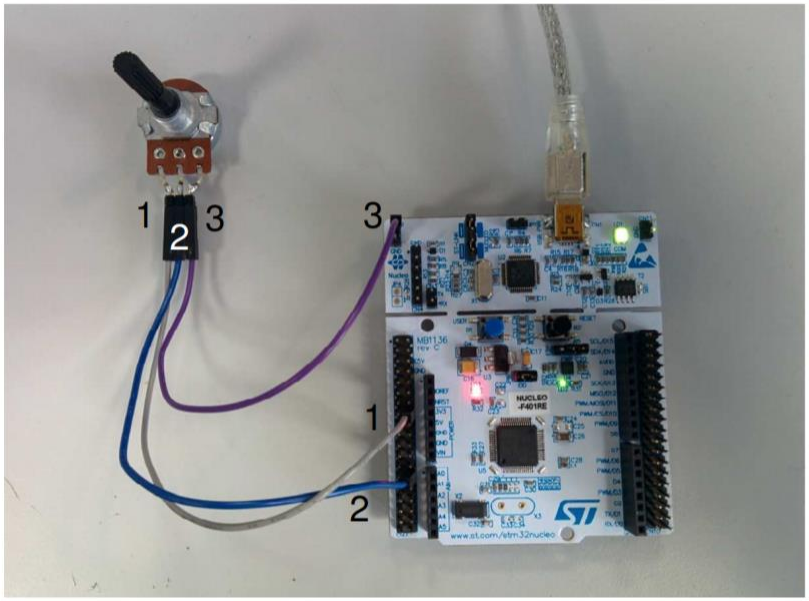



```
397 * @brief Function implementing the myTask02 thread.
398 * @param argument: Not used
399 * @retval None
400 */
401 /* USER CODE END Header_adcTask */
402 void adcTask(void *argument)
403 {
404   /* USER CODE BEGIN adcTask */
405   /* Infinite loop */
406   for(;;)
407   {
408
409         HAL_ADC_Start(&hadc1);
410
411         HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);
412
413         float rawValue = HAL_ADC_GetValue(&hadc1);
414         rawValue = ((float)rawValue) / 4095 * 10000;
415
416         htim2.Instance->CCR1 = (uint16_t) rawValue+100;
417
418         char msg[10];
419         sprintf(msg,"%d\n",(uint16_t)rawValue);
420         HAL_UART_Transmit(&huart2,msg, strlen(msg), 0xFFFF);
421
422   }
423   /* USER CODE END adcTask */
424 }
425
426 /**
427   * @brief  Period elapsed callback in non blocking mode
428   * @note   This function is called  when TIM1 interrupt took place, inside
429   * HAL_TIM_IRQHandler(). It makes a direct call to HAL_IncTick() to increment
430   * a global variable "uwTick" used as application time base.
431   * @param  htim : TIM handle
432   * @retval None
433   */
```

NUCLEO F401RE – ESEMPIO

USIAMO UN POTENZIOMETRO PER REGOLARE LA LUMINOSITÀ DI UN LED IN PWM

collegamenti



1. 3.3V => Pin 16 CN7
2. AIN0 => Pin 28 CN7
3. Ground