Register	Name	Description
GPIOx->MODER	Port mode register	These bits are written by software to configure the I/O direction mode. Input (reset state), general purpose output mode, alternate function mode and analog mode.
GPIOx->ODR	Port output data register	These bits can be read and written by software. This register controls pin output state (high/low).
GPIOx->IDR	Port input data register	Input pin state (high/low) can be read by using this register.
GPIOx->AFR[0]	Alternate function low register	Alternate function low selection for pins. Each pin has its own alternate function. System has 16 different alternate functions for pins.
GPIOx->AFR[1]	Alternate function high register	Alternate function high selection for pins. Each pin has its own alternate function. System has 16 different alternate functions for pins.
GPIOx->PUPDR	Port pull-up/pull-down register	Input pin no pull-up or pull-down, pull-up, pull-down selection.
GPIOx->OTYPER	Port output type register	Output configuratio for output pin(s). Output can be configure to push-pull or open-drain.
RCC->AHBENR	Peripheral clock enable register	Input/Output GPIOx clock enable.
RCC->APB1ENR	Peripheral clock enable register 1	Peripheral clock enable register 1. Enables for example USART 2-5, timers 2-7 and I2C1-2 etc…
USARTx->BRR	USARTx baud rate register	Baudrate must be calculated to this register in order to use USARTx.
USARTx->CR1	USARTx control register 1	Controls USARTx and for example this register determines word lenght.
USARTx->SR	USARTx status register	USARTx status. For example indication for transmit data register empty and indication when received data transferred to USARTx_DR register.
USARTx->DR	USARTx data register	Contains the Received or Transmitted data character, depending on whether it is read from or written to.
USARTx->CR2	USARTx control register 2	For example USARTx stop bit configuration.

SysTick->CTRL	Systick control and status register	This timer (systick) is dedicated to the OS, but could also be used as a standard downcounter.It is based on a 24-bit downcounter with autoreload capability and a programmable clock source. For example Systick counter enable and clock source selection etc.
SysTick->LOAD	SysTick reload value register	The LOAD register specifies the start value for the Systick downcounter.
SysTick->VAL	SysTick current value register	Current counter value for Systick timer.
RCC->APB2ENR	Peripheral clock enable register 2	For example timers 9-11 clocks can be enabled or USART1EN can be enabled.
TIMx->PSC	TIMx prescaler	Timer x prescaler (divider) for APB1 bus frequency.
TIMx->ARR	TIMx auto-reload register	For example in TIMx upcounting mode, the counter counts from 0 to the auto-reload value (content of the TIMx_ARR register), then restarts from 0 and generates a counter overflow (update) event.
TIMx->SR	TIMx status register	When an (overflow) update event occurs, all the registers are updated and the update flag (Update Interrupt Flag UIF bit in TIMx_SR register) is set.
TIMx->CR1	TIMx control register 1	For example selection for TIMx up/down mode and TIMx counter enable.
TIMx->CNT	TIMx counter	Current timer x counter value.
TIMx->CCMR1	TIMx capture/compare mode register 1	For example output pin Toggle can be selected in CCMR1 register. Toggle event when TIMx_CNT=TIMx_CCRx.
TIMx->CCR2	TIMx capture/compare register 2	Capture/Compare value. Can locate CCRx register depending output Channel (see in timer block diagram)
TIMx->CCER	TIMx capture/compare enable register	For example Capture/Compare 2 output enable. For example OC2 signal is output on the corresponding output pin. Corresponding output pin must be configured with MODER register alternate function.
TIMx->DIER	TIMx DMA/Interrupt enable register	Timer interrupt enable register.

void __disable_irq();	Disable interrupts	Disables all interrupts
void __enable_irq();	Enable interrupts	Enables all interrupts
SYSCFG->EXTICR[3]	External interrupt configuration register 4	This register bits are written by software to select the source input for the EXTIx external interrupt. For example port C (PC13) for EXTI13 can be selected "SYSCFG->EXTICR[3] |= 0x0020;". (access must be EXTICR[n-1] where n is register number)
EXTI->IMR	Interrupt mask register	Interrupts must be masked. For example in Nucleo board EXTI13 (MR13).
EXTI->FTSR	Falling edge trigger selection register	External interrupt falling edge selection.
EXTI->RTSR	Rising edge trigger selection register	External interrupt rising edge selection.
void NVIC_EnableIRQ(IRQn_t IRQn)	Enable IRQn (function)	Specified interrupt enable.
EXTI->PR	EXTI pending register	This register is software controlled and used interrupt corresbonding bit have to be cleared after each interrupt event.
ADC1->SMPR3	ADC sample time register 3	Sampling time can be selected. Reference manual page 280 shows sampling time versus resolution
ADC1->CR1	ADC control register 1	Bits 25:24 RES[1:0]: Resolution can be selected 6,8,10 or 12-bits.
ADC1->CR2	ADC control register 2	A/D Converter ON/OFF swich, mode selection, start conversion switch for regular channels and Data alignment selection.
ADC1->SR	ADC status register	ADC status can be read from this register.
ADC1->DR	ADC regular data register	These bits are read-only. They contain the conversion result from the regular channels.
ADC1->SQR5	ADC Regular sequence register 5	Select an ADC input channel. STM32L152RET6 datasheet pages 37-55 shows ADC_INx numbers. The temperature sensor is internally connected to the ADC_IN16 input channel which is used to convert the sensor output voltage into a digital value.

ADC->CCR	ADC common control register	For example internal temperature sensor activation.
DAC->CR	DAC control register	DAC channel enable and output buffer ON/OFF selection.
DAC->DHR12R1	DAC channel1 12-bit right-aligned data holding register	Software controlled DAC output register.
I2C1->CR1	I^2C Control register 1	I^2C Control register 1. For example start and stop bit generation.
I2C1->CR2	I^2C Control register 2	For example peripheral clock frequency have to be written to this register.
I2C1->CCR	I^2C Clock control register	I^2C bus frequency selection
I2C1->TRISE	I^2C TRISE register	I^2C bus risetime selection
I2C1->SR1	I^2C Status register 1	I^2C Status register 1. For example Address transmission indication can be found from SR1 and in 7-bit addressing, the SR1 register ADDR bit is set after the ACK of the byte. ADDR is not set after a NACK reception.
I2C1->SR2	I^2C Status register 2	I^2C Status register 2. SR2 is used for example to clear SR1 ADDR flag. ADDR have to be cleared in master mode by reading I2C_SR2 after reading I2C_SR1 which clears the SR1 register ADDR flag.
I2C1->DR	I^2C Data register	Byte received or to be transmitted to the bus.