<table>
<tr><th></th><th>STM32L1</th><th>STM32L4</th></tr>
<tr><td rowspan="3">GPIO</td><td>Default mode is Digital Input</td><td>Default mode is Analog</td></tr>
<tr><td>Alternative functions are different.

AF0 SYSTEM
AF1 TIM2
AF2 TIM3/TIM4/TIM5
AF3 TIM9/TIM10/TIM11
AF4 I2C1/I2C2
AF5 SPI1/SPI2
AF6 SPI3
AF7 USART1/ USART2/ USART3
AF8 UART4/UART5
AF9
AF10 USB
AF11 LCD
AF12 FSMC
AF13
AF14 RI
AF15 EVENTOUT</td><td>Alternative functions are different.

AF0 SYSTEM
AF1 TIM1/TIM2/TIM5/TIM8/LPTIM1
AF2 TIM1/TIM2/TIM3/TIM4/TIM5
AF3 TIM8
AF4 I2C1/I2C2/I2C3
AF5 SPI1/SPI2
AF6 SPI3/DFSDM
AF7 USART1/USART2/ USART3
AF8 UART4/UART5/LPUART1
AF9 CAN1/TSC
AF10 OTG_FS/QUADSPI
AF11 LCD
AF12 SDMMC1/COMP1/COMP2/FMC/SWPMI1
AF13 SAI1/SAI2
AF14 TIM2/TIM15/TIM16/TIM17/LPTIM2
AF15 EVENTOUT</td></tr>
<tr><td></td><td>Add a new register GPIO_ASCR (Analog Switch Control Register)

0: Disconnect analog switch to the ADC input
1: Connect analog switch to the ADC input

<pre>typedef struct {
  __IO uint32_t MODER;
  __IO uint32_t OTYPER;
  __IO uint32_t OSPEEDR;
  __IO uint32_t PUPDR;
  __IO uint32_t IDR;
  __IO uint32_t ODR;
  __IO uint32_t BSRR;
  __IO uint32_t LCKR;
  __IO uint32_t AFR[2];
  __IO uint32_t BRR;
  __IO uint32_t ASCR;
} GPIO_TypeDef;</pre>
For example, to use PA.2 as analog ADC input:
GPIOA->ASCR |= 1U<<2;</td></tr>
</table>