
```

typedef struct { GPIO_TypeDef *port;                                     gpio.h
      int           bit;
} Pin_t;

typedef enum {INPUT, OUTPUT, ALTFUNC, ANALOG} PinMode_t;
typedef enum {LOW, HIGH} PinState_t;
typedef enum {FALL, RISE} PinEdge_t;

void GPIO_Enable(Pin_t pin);
void GPIO_Mode(Pin_t pin, PinMode_t mode);
PinState_t GPIO_Input(Pin_t pin);
void GPIO_Output(Pin_t pin, PinState_t state);
void GPIO_Toggle(Pin_t pin);
void GPIO_Callback(Pin_t pin, void (*fn)(void), PinEdge_t edge);

```

```

typedef unsigned int Time_t;                                         systick.h

void StartSysTick();
Time_t TimeNow();
Time_t TimePassed(Time_t since);

```

```

typedef struct { TIM_TypeDef *iface;                                     timer.h
      int           chan;
      Pin_t          pin;    // Optional
      int           af;     // Optional
} TimerIO_t;

typedef enum {INCAP,      OUTCMP} TimerMode_t;
typedef enum {FALL, RISE, TOG, PWM1, PWM2} TimerSelect_t;
typedef enum {UP, CC1, CC2, CC3, CC4} TimerFlag_t;

void TimerEnable(TimerIO_t timer);
void TimerMode(TimerIO_t tio, TimerMode_t mode, TimerSelect_t sel);
void TimerPeriod(TimerIO_t tio, uint16_t psc, uint16_t arr, uint16_t rcr);
void TimerOutput(TimerIO_t tio, uint16_t ccr);
uint16_t TimerInput(TimerIO_t tio);
void TimerCallback(TimerIO_t tio, void (*fn)(void), TimerFlag_t flag);
void TimerStart(TimerIO_t tio);

```

```

typedef struct { ADC_TypeDef *iface;                                     adc.h
      int           chan;
      Pin_t          pin;    // Optional
} ADCInput_t;

void ADC_Enable(ADCInput_t ai);
uint32_t ADC_Read(ADCInput_t ai);

```
