8.1 Timers and time measurement

Timers and time measurement are important concepts in Arduino programming, as they allow you to perform tasks at specific intervals, measure elapsed time, or synchronize events. The Arduino has several built-in timer modules that you can use in your programs.

Here are some common ways to use timers and measure time in Arduino:

delay() function: This function causes the program to pause for a specific number of milliseconds. For example, the following code will cause the LED on digital pin 13 to blink every second:

```
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);  // wait for 1 second
  digitalWrite(13, LOW);
  delay(1000);  // wait for 1 second
}
```

millis() function: This function returns the number of milliseconds that have elapsed since the Arduino was powered on or reset. You can use this function to measure elapsed time or to trigger events at specific intervals. For example, the following code will turn the LED on and off every 5 seconds:

```
unsigned long previous_time = 0;  // store the previous time

void loop() {
  unsigned long current_time = millis();  // get the current time
  if (current_time - previous_time >= 5000) {  // check if 5 seconds have passed
    digitalWrite(13, !digitalRead(13));  // toggle the LED
    previous_time = current_time;  // update the previous time
  }
}
```

Hardware timers: The Arduino has several hardware timers that can be used to generate periodic interrupts. You can use these timers to trigger events at specific intervals without using the delay() function. For example, the following code uses Timer 1 to toggle the LED on and off every second:

```
void setup() {
  // set up Timer 1 to generate an interrupt every 1 second
  cli();  // disable global interrupts
  TCCR1A = 0;  // set Timer 1 to normal mode
```