



ARDUINO

interfacing techniques: timing of acquisition and control

© Nicola Lotti, Vittorio Sanguineti, Cecilia De Vicariis, 2018-2020. Materials provided "AS IS"; no representations or warranties provided. User assumes all responsibility for use, and all liability related thereto, and must independently review all materials for accuracy and efficacy.

Timed acquisition/actuation

- In many situations, need to acquire data and/or to control a device at a specified rate
- Simplest way to control rate is to use the delay() command within the loop() function:

```
Time = 1000; // Sampling period is 1000 ms ie 1 s

void setup(){}

void loop() {
  GetDataAndDoSomething(); // Acquire some Data and do something with it
  delay(Time); // wait some time
}

void GetDataAndDoSomething(){...}
```

- However, data acquisition and processing take time, so that the actual time period is slightly greater... (may even vary as computation time may not be constant)
- Not a big deal if the sampling period is much greater than the acquisition+processing time...
- **But what if you need to sample data real fast?**
Example: EMG signal needs a sampling rate of 2 kHz (period 0.5 ms)

Using timers

- **Solution:** use a 'timer' to trigger the acquisition. The timer should keep counting while acquisition proceeds

```
#include <FlexiTimer2.h>    //http://www.arduino.cc/playground/Main/FlexiTimer2

Time = 1; // Sampling period is 1 ms

void setup(){
  noInterrupts();          // Disable all interrupts before initialization
  FlexiTimer2::set(Time, GetDataAndDoSomething); // Attach function to timer
  FlexiTimer2::start();    // Start timer
}

void loop(){
  // empty! Acquisition if controlled by timer not the loop!
}
```

- Whenever the timer is reset, the data acquisition function is called... timer reset behaves very much like an interrupt
- **NB1: the function execution time has to be less than the sampling period!**