arm Education

Embedded Systems Essentials with Arm: Getting Started

Module 6

TP (6): Thinking point

Your alarm clock wakes you in the morning, it's 7 am. You get up and go out for your morning jog. You run a fixed route, as fast as you can. You set your watch on stopwatch and stop it the moment you're back home. After a shower you go to the kitchen to make toast – you're very particular, the toast must be on exactly 3 minutes, not more, not less. You eat quickly, as your tram for work leaves at 8.15. You catch it just in time. Once in your seat you sit back and admire how its electric motor pulls you away so smoothly.

Your day has just begun, but already you've used time in many different ways. Importantly, you've used time in ways similar to how a microcontroller uses it. Some of these ways are familiar and visible, others you may not even recognize.

All day long you're conscious of time, passing continuously, available to you from any clock or timepiece. (Sometimes we call this a **Real Time Clock.**) But you've set your alarm clock to a particular time, 7 am, and all through the night your clock is checking whether that time has been reached. It's **comparing** the present time with your alarm time. When they're the same – at 7am – it causes an event, the alarm goes off!

You go out for the jog. You've set your stopwatch to zero and started running. As you finish the jog, you stop the watch; in doing so you **capture** the time as the run finishes. By looking at that information, you know how long you were running for. Or you can use a timer in a different way, to time a stand-alone activity which must be of fixed duration. Don't burn that toast now, will you?!

Time, and how we deal with it, is so very important in the embedded world. We're going to see examples of timing, capturing and comparing from the microprocessor and Mbed world. A lot of it is going to feel like we're just using stopwatches and alarm clocks in a slightly different way.

But how about the last point mentioned in your early morning schedule? What has all this got to do with the motor on the electric tram, and the way it's controlled? (Let's assume for now that this is a modern tram, recognizing that there are some pretty old ones around as well.) Well, microcontrollers need to control a huge variety of things which are analogue in nature, anything from the brightness of a humble LED, to the speed of a powerful motor that drives a tram or electrical car. But the signals flying around in a microcontroller are more or less all digital, on their own they can just switch a motor on or off. A very clever way of using time to exercise variable control, which you're going to see, is called **Pulse Width Modulation.** By controlling the pulse width of a stream of digital pulses, and then varying as needed, we can get that digital microcontroller to control those analogue things. Yes, we might need a bit of power electronics in between, but we are here opening the door to one of the great techniques of computer control of physical actuators.

Enjoy your Module 6, and take time to understand all the important, and sometimes challenging ideas it contains!