Microprocessor Applications I

Supplementary Lecture 02

Authors:
Chris Crary
Wes Piard

Lecture Overview

1. Present problem somewhat similar to Lab 2
   - Discuss problem and example data
2. Solve problem
   - Pose questions
   - Answer questions
   - Explain pseudocode/flowchart for design
   - Briefly explore Atmel datasheets
   - Write code (not everything will be given)
   - Discuss more concepts
   - Demo completed program
3. Answer any questions outstanding

Please note: Due to time allotment, it is quite possible that not all planned points will be discussed during this lecture.

Problem

You must create and compile an assembly program on your Atmel XMEGA128A1U microprocessor to continuously blink the LEDs on your *Switch & LED Backpack* at a rate of 5Hz while tactile switch S1 is held down. In other words, when tactile switch S1 is not held down, your LEDs will remain OFF.

Questions To First Pose

1. Where are the switches and LEDs?
2. What are the switches/LEDs connected to on the microprocessor?
3. What are I/O ports?

I/O Ports

- In the context of this course, an **I/O port** can simply be related to a collection of physical pins that serve as either “inputs” or “outputs”.

- **The motivation for pins within an I/O port is to allow communication between the microprocessor and internal/external hardware**

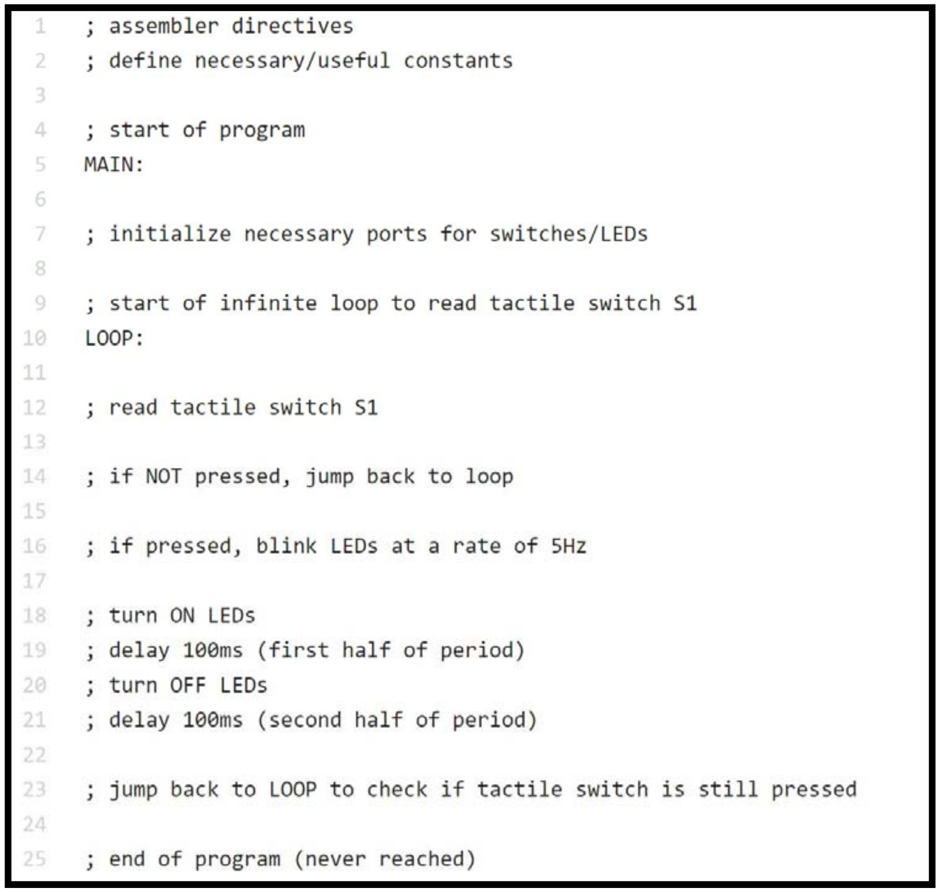
- These pins hold an electrical voltage level, either a “high” voltage (also known as ‘1’), or a “low” voltage (also known as ‘0’).

- In our processor, there are many I/O ports, e.g., PORTA, PORTB, PORTC, …

- The current “value” of each pin within a port (either ‘0’ or ‘1’) is stored within a specific bit of a predefined XMEGA *data* memory location, specified in the include file (see *PORTX_IN*).

- There exist registers in the processor that allow you to access information about a port or configure a port, e.g., there exists a register that allows you to configure a port’s pins to serve as inputs or outputs (see *PORTX_DIR, PORTX_DIRSET, PORTX_DIRCLR)*, etc.

Pseudocode & Flowchart



```
; assembler directives
; define necessary/useful constants

; start of program
MAIN:

; initialize necessary ports for switches/LEDs

; start of infinite loop to read tactile switch S1
LOOP:

; read tactile switch S1

; if NOT pressed, jump back to loop

; if pressed, blink LEDs at a rate of 5Hz

; turn ON LEDs
; delay 100ms (first half of period)
; turn OFF LEDs
; delay 100ms (second half of period)

; jump back to LOOP to check if tactile switch is still pressed

; end of program (never reached)
```

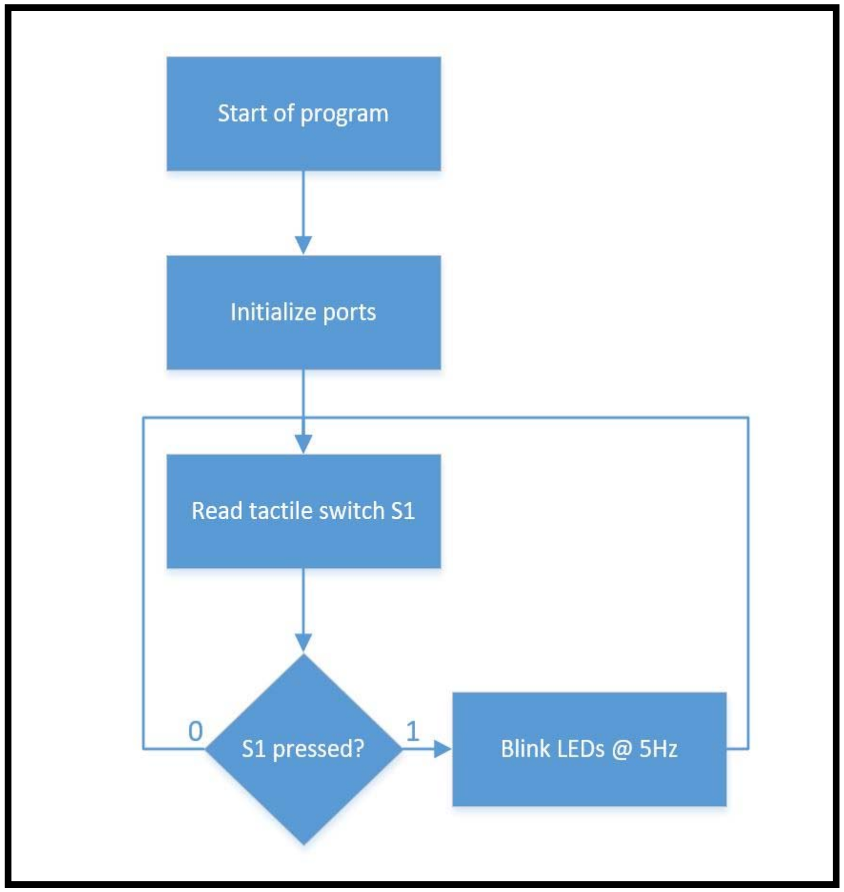


Figure 1: Pseudocode (left) and flowchart (right) for given problem

Exploring The Atmel Datasheets

- See Dr. Schwartz' website: https://mil.ufl.edu/3744/
    - Navigate to the **Atmel AVR XMEGA** section of the *Software/Docs* webpage

Time To Code

Doin’ it live

Questions?