Embedded Systems Mini-Project (0.25 Bonus Points)

D-ITET
Center for Project-Based Learning

PD. Dr. Michele Magno

December 9, 2024

1 Objective

The goal of this individual mini project is to allow students to apply their acquired knowledge of embedded systems to realize a small fully working application with the reference STM32 student platform used throughout the course. The project involves implementing data readout and processing for a selected sensor, implementing the signal processing code on the microcontroller and profiling the performance, as well as analyzing energy consumption.

2 Project Scope

Within the scope of the mini-project, you will **realize a real-world application**. For this, choose one of the following applications and implement it during your mini-project:

- **Fitness Tracking Application (IMU sensor):** The STM32 student platform is programmed to be used as wearable device that tracks user activity (i.e detect walking, standing, running on the spot).
- **Gesture Recognition System (IMU sensor):** Use the IMU to detect hand gestures (left, right etc.). This can be applied to control devices or applications based on gestures.
- **Tilt Sensing Application (IMU sensor):** Create a tilt sensor that detects the tilt angle of a surface. This can be used in applications such as leveling tools or as a part of a mobile game.
- **Acoustic Sensor for Intruder Detection (microphone):** An embedded system that uses a microphone to detect unusual sounds, such as glass breaking, footsteps, or door reaches, to trigger a security alert.
- **Noise-Level Monitoring and Alert System (microphone):** A system designed to monitor ambient noise levels in environments like factories, schools, or hospitals and alert when noise exceeds safe thresholds.
- **Your own idea:** an application based on the IMU or microphone, that follows the outline of the mini-project given below.

Once you have selected the application, the following is expected from the mini-project:

1. As all of the applications are energy sensitive, you will implement two modes in your application: the active mode and sleep mode. All processing and sensor data collection will be done in real-time on the STM32.
    - **Active mode:** during this mode you will perform the data collection from the sensor, processing the sensor values and create an actuation. For example:

  - Collect data from the IMU sensor and select a fixed sample rate (i.e 100Hz) and period (i.e. 1second of data)
  - Process the data to evaluate the content of the data (i.e in use-case 1 decide if the data stream is a walking standing , running, ...). You will implement at least two different processing algorithms for later comparison. For example: (i) Compute the average of accelerometer values using 8-bit integer data and (ii) perform a simple frequency analysis of the accelerometer data using 32-bit floating-point values.
  - Display the results via LED and/or via UART on your PC's terminal
- **Sleep Mode:** is entered after finishing the active mode and will last for exactly 2 seconds, before the active mode restarts.

2. Once you have a basic working code for your application, the next task is profiling its performance with respect to clock cycles needed.
   - Measure the number of operations, respective clock cycles, which are needed to execute the data processing task. This profiling is done for each of the at least 2 different algorithms you implemented.
   - Record the clock frequency used during the experiment (80 MHz or another suitable frequency).
3. Finally, based on the profiling done, estimate the energy consumption of your application. Assume the system runs at the selected clock frequency in the active state.
   - Assume 1mW per MHz for the active state and 10uW for the sleep state.
   - Calculate and compare the energy used during active processing (individually for all implemented algorithms) vs. in the energy in (sleep) mode. How do they compare to each other?

3 Deliverables

- **Report (maximum half page A4, 12pt):** as .pdf file. The report will cover:
  - The project goal and sensor selected, with a focus on its real-world application.
  - Chosen processing algorithms and reasoning why.
  - The number of operations per clock cycle achieved with the two implemented processing tasks and discussion if it can be improved (e.g. by using CMSIS DSP)
  - Evaluation of energy consumption, comparing the energy for active mode and sleep mode, as well as for the two processing tasks.
- **Video of the demo** showing it in operation and interaction with the system via the console (max. 1 minute).
- **Source code of the demo** as single .zip archive file.

4 Evaluation Criteria for the Bonus (0.25 points)

To receive the bonus, students must:

- Collect data from at least one sensor and process it using a fixed sample rate and period.
- Provide a clear performance comparison of the 2 algorithms implemented in terms of speed and energy consumption.
- Include energy consumption calculations based on active and sleep phases.
- Submit a concise, half-page report summarizing their findings.
- Submit a 30sec to max. 1 minute video of the developed application, showing it's correct operation.
- Submit the code used to create the results.

all deliverables must be handed in via Moodle until Dec. 20th 2024, 23:59 CET.

5 Support

The mini-project should showcase the student's own capability in the area of embedded systems. However, for concise questions or issues the students may refer to the teaching assistants during the regular embedded systems exercise hours (no support via Moodle, and no support in the office hour).