Pedometer for STM32F4
Use the MEMS accelerometer to count number of steps

De Donatis Emanuele
Matr. 817667, (emanuele.dedonatis@mail.polimi.it)
Pistone Bruno
Matr. 818949, (bruno.pistone@mail.polimi.it)

Report for the master course of Embedded Systems

July, 18 2014

Abstract

This project is part of a collaborative project whose purpose is to develop a simple yet realistic training support system. The goal of this project is to develop a pedometer using the MEMS motion sensor on the STM32F4DISCOVERY board. The device provides the user with statistics about its training activity.

1 Introduction

The Real-Time Operating System course of the Polytechnic of Milan proposed a collaborative project whose aim is to develop a Personal Trainer device based on the economical development board STM32F4DISCOVERY and the Miosix embedded OS.

Module Name	Device
Pedometer	MEMS
User Interface	UART, Flash
Audio Feedback	CS43L22 (DAC)
Voice Commands	NONE (PCM Samples)
PCM Encoding	MP45DT02 MEMS Mic
Social Wireless	NRF24L01 2.4GHz TxRx
Context Awareness	ADC

Table 1: Group modules

The final goal is to join each module in order to build a working system.

1.1 STM32F4DISCOVERY board

The STM32F4DISCOVERY is an evaluation board by STMicroelectronics, based on the ARM Cortex-M4F core.

- STM32F407VGT6 microcontroller featuring 32-bit ARM Cortex-M4F core, 1 MB Flash, 192 KB RAM in an LQFP100 package
- LIS302DL or LIS3DSH ST MEMS 3-axis accelerometer
- MP45DT02, ST MEMS audio sensor, omni-directional digital microphone
- CS43L22, audio DAC with integrated class D speaker driver
- USB OTG FS with micro-AB connector

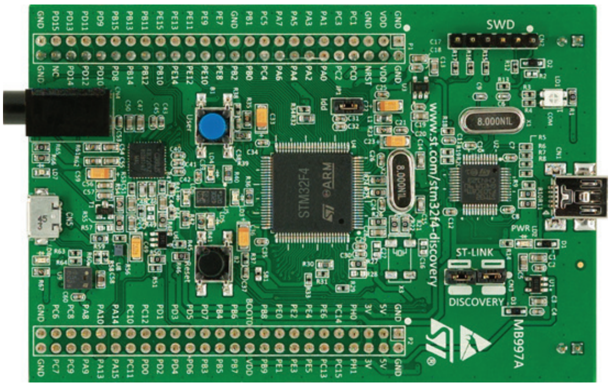


Figure 1: STM32F4DISCOVERY board

1.2 MIOSIX embedded OS

Miosix is an OS kernel designed to run on 32bit microcontrollers, in active development since 2008. It supports both a single process, multiple threads application model where applications are statically linked with the kernel, and an experimental multiprocess environment with memory protection that allows loading applications at runtime. The kernel is royalty-free and licensed under the GPL license with an exception that allows it to be linked with propietary application code. [1]

1.3 MEMS Motion Sensor

The STM32F4DISCOVERY includes the LIS302DL ST MEMS motion sensor. It is an ultra compact low-power three-axis linear accelerometer. It includes a sensing element and an IC interface able to provide the measured acceleration to the external world through I2C/SPI serial interface.

The STM32F4 controls this motion sensor through the SPI interface (see **Figure 2**).

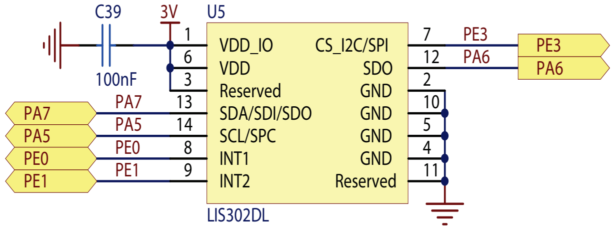


Figure 2: LIS302DL Motion Sensor

2 Pedometer Algorithm

To develop the pedometer algorithm we relied on the Neil Zhao's model .[4]

2.1 Digital Filter

A digital filter is needed to smooth the signals. Four registers and a summing unit can be used (see **Figure 3**). More registers could be used to make the acceleration data smoother, but the response time would be slower.

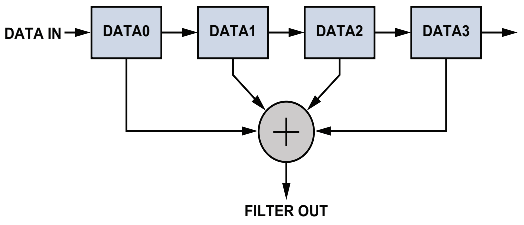


Figure 3: Digital filter

2.2 Dynamic Threshold and Dynamic Precision

The system updates the maximum and minimum values of the 3-axis acceleration every 50 samples. The average value, (Max + Min)/2, is called the dynamic threshold level.

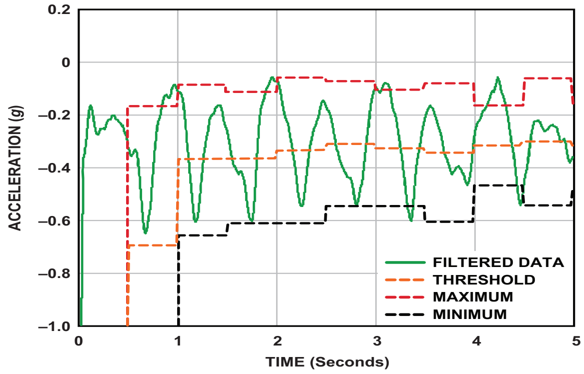


Figure 4: Acceleration curves

For each axis two registers, sample_new and sample_old, are used to pick out new sample result. When a new data sample comes, sample_new is shifted to the sample_old register unconditionally. However the new sample result will be shifted into the sample_new register only if the changes in acceleration are greater than a predefined precision; otherwise the sample_new register will remain unchanged. The shift register group can thus remove the high-frequency noise and make the decision more precise.

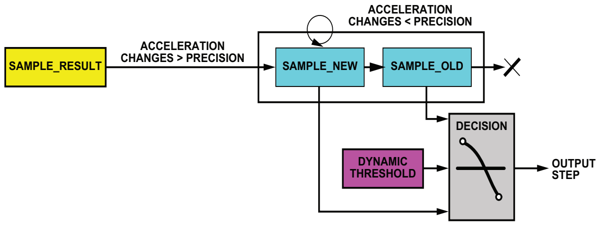


Figure 5: Dynamic threshold and dynamic precision

2.3 Step

A step is defined as happening if there is a negative slope of the acceleration plot (sample_new < sample_old) when the vertical acceleration curve crosses below the dynamic threshold.

2.4 Peak Detection

The step counter calculates the steps from the x-axis, y-axis, or z-axis, depending on which axis has the maximum absolute value of threshold, since the gravitational force acts on it.

2.5 Time Window

Time window is used to discard the invalid vibrations. We assume that people can run as rapidly as five steps per second and walk as slowly as one step every two seconds. Thus, the interval between two valid steps is defined as being in the time window [0.2 s to 2.0 s]; all steps with intervals outside the time window should be discarded.

3 Statistics

3.1 Training mode

It is possible to recognize the training mode based on the step's frequency: the running frequency is between 2 and 5 steps per second and the walking frequency is between 0.5 and 2 steps per second.

Minimum	**Mode**	**Maximum**
200ms	RUNNING	500ms
500ms	WALKING	2000ms
2000ms	STEADY	∞

Table 2: Interval time between two steps

3.2 Distance

The distance depends on the length of a step. Experimentally we can approximate this length as:

$$walking\_length = 1/6 * height$$
$$running\_length = 1/3 * height$$

3.3 Speed

Knowing the distance and the time between two steps we can calculate the approximate speed:

$$speed = distance/time$$

3.4 Calories

There is no accurate means for calculating the rate of expending calories. Some factors that determine it include body weight, intensity of workout, conditioning level, and metabolism. We can estimate it using a conventional approximation:

$$calories[kCal] = 0.9 * distance[km] * weight[kg]$$

4 User Interface

4.1 LCD HD44780

We used an LCD HD44780 20x4 display to allow the user to view their own statistics. Miosix provides a C++ library to easy use the LCD.

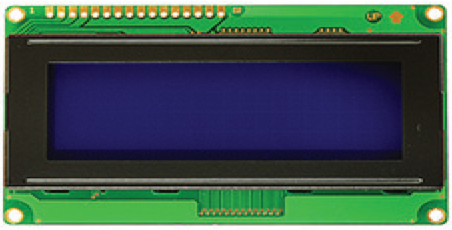


Figure 6: LCD HD44780 20x4

The user can view the steps, the training mode, the instantaneous speed, the distance and the burned calories.

S	T	E	P	S	:		*	*	*	*									
M	O	D	E	:		*	*	*	*	*	*								
S	P	E	E	D	:		*	*	.	*		k	m	/	h				
K	M	:		*	*	.	*	*		K	C	A	L	:		*	*	*	*

Table 3: GUI tamplate

4.2 User buttons

Some stats, like distance, speed and burned calories, required user's weight and height. On startup the system asks for this information and through three buttons the user can increase / decrease / confirm its weight and height.

5 Social Wireless

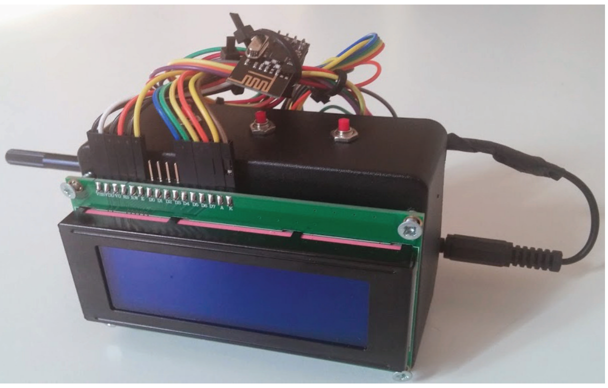
We collaborate with the group 61 to implement the wireless module. [3]
The goal is to exchange data with nearby devices using a NRF24L01 2.4GHz RF Transceiver.
The system sends and receives the number of steps in order to compare own stats with friends' stats.

6 Audio Feedback

We collaborate with the group 31 to implement the audio module. [2]
The goal is to notify the user about his number of steps and the result of steps' comparison with his friends.

7 Conclusion

This is the final prototype. On the left there is a potentiometer to adjust the contrast of the LCD. On the top there are the three user buttons and the NRF24L01 transceiver. On the right there is the audio jack connected to a mini-speaker. We used a portable battery charger for smartphone to power the device via USB.



References

[1] http://miosix.org/.

[2] https://github.com/derkling/rtos2013/tree/grp31.

[3] https://github.com/derkling/rtos2013/tree/grp61.

[4] Neil Zhao. Full-featured pedometer design realized with 3-axis digital accelerometer.