Introduction
Introduction

- As more and more sensors are incorporated into portable electronics such as smartphones and wearable devices, “Sensor signal processing algorithm” is becoming just important as the sensors.

- MegaChips’ frizz, with its 32bit DSP based motion engine “ParaForce”, can realize high performance calculations used in processing algorithms with ultra low power consumption in lieu of microprocessor which often runs at hundreds of MHz and consumes much higher power, thereby extending the runtime of portable devices.

- To facilitate faster development cycle, MegaChips can provide algorithm for indoor navigation, which utilizes the special functions from ParaForce engine that’s inside frizz. (PDR library for frizz)
Introduction to frizz
【Motion Engine & Sensor–HUB IC】

- Features
- Application examples
- Specification
- Comparison with competitors
- Roadmap plan
- frizz is an ultra compact LSI with embedded 32bit DSP based motion engine “ParaForce”
- frizz is designed to perform high level calculations at very low clock speed and, therefore, require very little power, making it ideal for portable electronics such as smartphones and wearable devices.
### Features

- **frizz** includes “ParaForce”, which is based on Tensilica 32bit DSP Xtensa LX4 enhanced by MegaChips.
  - 3way VLIW
  - Floating Point 4way SIMD

- **frizz** can perform high level calculations as required by PDR at low clock speed and consume very little power.
  *PDR (Pedestrian Dead-Reckoning) for more information refer “PDR library for frizz”

- **frizz** supports popular Eclipse based IDE for software development environment
  *for more information refer “Development environment”

- **frizz** can be used as a sensor–HUB LSI.
  Dedicated sensor–HUB can offload from host processor and help to reduce power consumption of entire system.
Application example 【Smartphone】

- frizz can take over the functions of sensor calibration, and data buffering from the host processor, thereby saving power, or freeing up the host processor to perform other functions.
- frizz and its software library for PDR, Context awareness and Activity monitoring provide the foundation to create new applications.
Application example 【Wearable devices】

- Frizz can recognize a person’s status such as walking, or running, estimate walking speed and calculate distance traveled. These are all the essential functions to make wearable devices stand out.
## Specification

<table>
<thead>
<tr>
<th><strong>CPU</strong></th>
<th><strong>Power Consumption</strong></th>
</tr>
</thead>
</table>
| 32bit DSP Motion engine “ParaForce”  
Max. Frequency 40MHz  
*Tensilica Xtensa LX4 base  
- 3way VLIW  
- Floating Point 4way SIMD | Active: 8.3mA@40MHz  
Sleep(fast): 1mA@40MHz  
Sleep(slow): 100uA@100kHz  
Stop: 100uA@100kHz  
Standby: 8.3uA |

<table>
<thead>
<tr>
<th><strong>RAM</strong></th>
<th><strong>External interface</strong></th>
</tr>
</thead>
</table>
| Instruction RAM: 256KB  
Data RAM: 256KB | Host CPU Interface: SPI or I2C  
Dual Purpose RAM (32bit x 64word) |

<table>
<thead>
<tr>
<th><strong>Power voltage</strong></th>
<th><strong>Device interface</strong></th>
</tr>
</thead>
</table>
| Core 1.2V  
IO 1.8/2.5/2.8/3.3V | UART x1, SPI x1 (4 devices control)  
I2C x1, GPIO x4 |

<table>
<thead>
<tr>
<th><strong>Power save mode</strong></th>
<th><strong>Package</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep/Stop/Standby mode</td>
<td>3.5mm x 3.5mm x 0.65mm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>System</strong></th>
</tr>
</thead>
</table>
| • Timer 32bit x4  
• Dual Purpose RAM(32bitx64Word) : FIFO mode or Snap Shot mode  
• Self Boot from SPI Flash memory  
• Host CPU Download boot  
• JTAG debug |

*Red characters for remarkable points

*Specification may change without any notices. Please ask sales representative for latest specification.*
**Performance comparison with competitors**

- The chart below compares the ability of the IP cores to perform matrix operation (from 4x4 to 16x16), which is used extensively for PDR, at the same 40MHz frequency.
- Frizz, with its support for floating point 4way SIMD instruction, can achieve over 4.4x performance when compared with Cortex-M4F based controller.

<table>
<thead>
<tr>
<th></th>
<th>4 × 4 Matrix Multiplication</th>
<th>12 × 12 Matrix Multiplication</th>
<th>16 × 16 Matrix Multiplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM Cortex-M4</td>
<td>309,598</td>
<td>20,564</td>
<td>9,402</td>
</tr>
<tr>
<td>Frizz</td>
<td>1,346,801</td>
<td>83,875</td>
<td>36,106</td>
</tr>
</tbody>
</table>

[Unit: Times]

- About 4.4x performance
- About 4.1x performance
**Power comparison with competitor**

- For PDR application, frizz can reduce up to **93%** power consumption compared with ARM Cortex-M4F processor!

*PDR (Pedestrian Dead-Reckoning) for more information refer “PDR Library for frizz”

- **This Chart** indicate min. clock speed for PDR calculation at 100Hz sampling rate.
  - ARM Cortex-M4F MCU: about 80MHz
  - frizz: 6.7MHz

- **Vertical axis** show power consumption for PDR calculation.
**Comparison with other IC【FFT arithmetic processing benchmark】**

The followings are the comparison data of frizz with ARM Cortex-M4 on FFT (1024point) arithmetic capacity used for analytical processing, voice processing and noise rejection processing for bio-sensing in frequency domain of sensor data.

<table>
<thead>
<tr>
<th></th>
<th>Execution cycle</th>
<th>Power consumption per one time execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM Cortex-M4</td>
<td>133.6K</td>
<td>57.1μW</td>
</tr>
<tr>
<td><strong>frizz</strong></td>
<td>54.6K</td>
<td>13.7μW</td>
</tr>
</tbody>
</table>

*76% decrease in power consumption compared to ARM Cortex-M4!*
### Roadmap plan

#### frizz Roadmap

<table>
<thead>
<tr>
<th>Year 2014</th>
<th>Year 2015</th>
<th>Year 2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>frizz</td>
<td>frizz2</td>
<td>frizz2 MCP</td>
</tr>
</tbody>
</table>

- PDR [Dead Reckoning Library]
- PDR+ [Improve precision combining WiFi positioning and Beacon positioning (BLE)]

#### Software Roadmap

- Motion recognition
- Pulse check
- Voice command
- Improved Voice command for Frizz2
Introduction
PDR library for frizz

- Main features
- PDR library structure
- Comparison with other indoor positioning system.
- Application example
The requirement for indoor positioning, where GPS signals cannot reach, is becoming more important for users, retailers, and app developers alike.

Smartphone and wearable devices are in need of technology that can estimate relative position, and monitor activities with context awareness by processing data from Accelerometer, Gyroscope and Geomagnetic sensors.

MegaChips developed PDR (PDR: Pedestrian Dead-Reckoning) library for frizz based on high precision algorithm developed by AIST (National Institute of Advanced Industrial Science and Technology).

MegaChips PDR library for frizz realizes advanced algorithm calculation in high speed with ultra low power consumption. The performance can help clients to develop applications such as indoor navigation or create new portable devices.
What is PDR (Pedestrian Dead Reckoning)?

- PDR is a system which estimates “Speed”, and “Direction” as well as position of target person by using relative movement data of inertial sensors such as acceleration, gyroscope and geomagnetic sensors.

- PDR is used for indoor navigation system, where GPS signals are unreliable or non-existent.
PDR library architecture and elemental technology

- **Altitude estimation**
  High precision estimation for altitude of target devise by using Kalman filter.

- **Walking motion detection**
  Judge walking motion from relationship between vertical and moving direction’s acceleration.

- **Walking speed estimation**
  Estimate walking speed from vertical acceleration.

- **Moving direction estimation**
  Estimate moving direction from geomagnetic.

- **Geomagnetic verification**
  Judge reliability of geomagnetic sensor.
PDR library application example1 (Motion recognition)

By using elemental technology of PDR library, it is possible to recognize human activity.

Walking

Running

Up or down stairway

Riding bicycle

In a car

On a train
PDR library application example2 (Context awareness)

- Estimate what target person is doing while walking.

- Swing arms
- Looking at device
- Detect state of holding devices
- Talking on the phone
- In pocket
Evaluation results of PDR library

- Evaluate walking 50m square in counterclockwise direction with three different way of holding device.

![Walking route](image)

<table>
<thead>
<tr>
<th></th>
<th>In pocket</th>
<th>Hold in hand</th>
<th>Swing arms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results</td>
<td>On the right track.</td>
<td>Step off the track compared to In pocket.</td>
<td>Deviate substantially from track compared to other 2 patterns.</td>
</tr>
</tbody>
</table>

▼ Results
## Application example 1

- PDR + Beacon for accurate indoor navigation

In the case of using PDR alone for indoor navigation, relative error will accumulate over time. If PDR is applied together with “Beacon”, which sends absolute position, and “Map matching technology, the accuracy of indoor navigation system can be greatly improved.

*Beacon is a information supplying system, which is supported from iOS7, applying “Bluetooth Low Energy” technology.

Cite from: EstimoteBeacon (http://estimote.com/)，Aplix (http://www.aplix.co.jp/?page_id=7593)
Application example 2

- Fitness application

frizz and its library can recognize and indicate type of motion (Run/Walk), speed, and distance traveled of target person.

- Healthcare application

[Example as a kick device]

Recognizing sleeping status (REM / Non-REM), and kick (control communication) to Alarm clock and lighting for optimal refreshing wake up.
Application example 3

Running coach

- User interface
- Target speed alert w/Statistical info.
- Arm swing correction alert

Search lost children indoors (@Supermarket)

- Where is he/she?
- BLE-equipped extension
- PDR positioning
- Confirm relative position between base unit/extension
- Confirm relative position
- BLE-equipped base unit
- PDR positioning

Watch over the elderly (@house/nursing homes)

- BLE transmitter
- Detect abnormal posture and perform protective action
- Detect abnormal posture and activate alarm
Development tool

- Software development tool
- Reference software
- Evaluation board
Software development environment

- Eclipse based IDE
- Based on GCC4.2.0
- Support C++
  Also support C99
- Standard C library (Newlib)
- HW accelerators can be called by intrinsic function.
  - Single precision floating point type
  - Single precision floating point X 4 type

For above types user can use following arithmetic operators.
  - Single line operator(-)
  - Binary operator(+-*)
  - Relational operator(<,<=,==,>=,!,=)\(^{(RE)}\)
Structure of reference software

Android on host processor

- User Android Application
- Android Frameworks
- NDK for PDR
- Sensor HAL
- Linux Kernel Driver for frizz

frizz firmware

- Peripheral Driver
- Sensor Driver
- I/O HAL
- Sensor HUB Framework
- Rotation
- Quaternion
- Step Counter
- PDR
- Context Rec.
- Activity Rec.

MegaChips can provide sample code for Android OS

- NDK for PDR
- Android Sensor HAL
- Linux Driver
- User Android Application

MegaChips can provide sample code for frizz PDR library.

- PDR library
- Context Recognition
- Activity Recognition

Scheduled development for frizz

- Additional Sample Code for libraries
- Additional sensor application
Evaluation system with BOSCH

“Chignon B” Evaluation board for wearable devices

- Small external size (20x40mm) for wearable devices
- Battery operation for demonstration and field test
- OLED (Organic LED) can be connected [Option]
- GPS module can be connected [Option]
- Debug board for software development is available

Sensor
Accelerometer/Gyro: Bosch BMI160
Magnetometer: Bosch BMM150
Pressure/Humidity/Temperature: Bosch BME280

Bluetooth Low Energy Module
MBH7BLZ02

GPS Module [Option: TBD]
GP-2106

OLED Module [Option: TBD]
DD-12832YW-1A