# sensing the FUTURE

**InvenSense** Developers Conference 2016



# SensorStudio

#### Introduction



### IDC - SensorStudio track - Agenda

- 1. Introduction Herve Blanc
  - The lot market expansion is calling for more flexibility. In this session, you can learn how SensorStudio will help you tackle the challenges ahead. Market trend, SensorStudio solution, and roadmap will be introduced.
- 2. New Features/Benefits Sheena Shi, Maaz Qazi, Christopher Rumpf
  - We are listening to you and a lot of effort has been put into addressing your feedback. Come learn about SensorStudio new release features and new development kits.
- 3. Deep Dive Herve Blanc
  - SensorStudio development environment will be detailed. You will learn how to add your sensor driver and create your algorithms, while benefiting from a complete sensor development framework.
- 4. Real Use cases Andreas Brauchli, Herve Blanc
  - We will review how Sensirion partner integrated its new SHT3x humidity sensor in SensorStudio.
  - We will discuss how we have used SensorStudio to put together prototypes, pulled new algorithms and demos together in no time.
- 5. Hands on Christopher Rumpf
  - In this session, you will be able to access the SensorStudio development kits, evaluate the software and ask questions to our developers.

### Foreword

# sensing the **FUTURE**

 "And we've analyzed over 700 swimmers, different body types, different abilities. We hooked them up to state-of-theart metabolic equipment. We've even drawn blood samples to look at lactic acid levels and we used all this body of information to create an algorithm that will give you the most accurate calorie burn information while you're swimming."

Source <a href="http://www.singjupost.com/apple-iphone-7-keynote-september-2016-launch-event-full-transcript/4/">http://www.singjupost.com/apple-iphone-7-keynote-september-2016-launch-event-full-transcript/4/</a>



Source <a href="http://www.apple.com/newsroom/2016/09/apple-introduces-apple-watch-series-2.html">http://www.apple.com/newsroom/2016/09/apple-introduces-apple-watch-series-2.html</a>



### • SensorStudio,

Motto

The sensor platform for the "Internet of Sensors", Simplifies software & algorithm development and Accelerate your time to market



### Why: Problem being solved

# sensing the **FUTURE**

SensorStudio is reducing Software Development cost/effort as:

- Developers can get started with sensing project quickly without prior sensing knowledge
- Developers can focus on combining sensors & algos
- Developers don't need to learn FreeRTOS configuration
- Sensor framework handle data synchronization
- Higher level of abstraction through visual programming
- Gain data insight through real time scientific visualizations
- Record & replay for a complete algo dev/validation cycle and paving the way to machine learning (e.g. TensorFlow)
- Creates compelling PoCs & demos in no time (get funding!)
- Customer final product works with SensorStudio too (all necessary source code is provided to support customization)

#### Overview

# sensing the **FUTURE**

"Flows" In graphics format Drag and drop elements, traces, inputs and outputs "Flow" in list format (does not show connections and dependencies)

В "Pool"

Collection of all building Blocks available for the flow design

"Console" window for Debug messages

Е

"Properties" Parameters of each element in Flow



InvenSense

### **Development Environment**

# sensing the **FUTURE**

SensorStudio SW

Nucleo HW



USB







- <u>SensorStudio</u> Software tool on PC
- CortexM4 sensor framework loads as a primitive in SensorStudio.
- Access to framework features and functions
- Custom algo development capabilities in C.
- Debug code with Eclipse
- <u>STm32 Nucleo</u> CortexM4 microcontroller
- "bridge mode" (used as interface card between SensorStudio and sensor framework)
- Load Final project from SensorStudio into Nucleo FLASH memory
- Run algo in Nucleo and SensorStudio simultaneously -> iterative development
- Debug code in Eclipse C/C++ (could be STm32 Cube)
- <u>InvenSense Shield</u> with ICM-20690, Mag, Pressure, Proximity

CM-20690 HW

### **GenericSensorHub on Nucleo**

- ✓ Robust sensor framework
- ✓ Rapid implementation
- ✓ Optimized for low-power
- ✓ Extensible for any sensor
- ✓ Host AP / MCU agnostic
- ✓ SOC is managed through a race-free protocol



### **Software Features**



- Sensor features supported on Cortex M4 MCU
- All features accessible via API from host application processor
- Feature running on Cortex M4 and power/code optimized
  - Motion
    - Calibrated Acc
    - Calibrated Gyro
    - Calibrated Mag
    - Linear Acceleration
    - Gravity
  - Orientation
    - Game Rotation Vector (AG-Orientation)
    - Rotation Vector (AGM-Orientation)
    - Orientation (AGM-orientation reported as yaw/pitch/roll angle)
    - Geomag Rotation Vector (AM-Orientation)
  - Gestures
    - SMD (Significant Motion Detector)
    - Pickup
    - Tilt
  - Activity Classification
    - Step Detector
    - Step Counter
    - Basic Activity Classifier (Still, Walk, Run, Biking)
  - Pressure BMP280
  - Proximity VCNL4040



InvenSense

### Visualize all signals Real-Time

- Different visualizations types
- InvenSense sensors & sensor fusion, your own!

sensing the **FUTURE** 

InvenSense



### Easily add sensors & algorithms

- Don't worry about RTOS
- Data synchronization is taken care off
- Focus on your new sensors or algorithms!

DoorOpeningDetection.ImplCode - C Editor		
API Documentation Search Replace Line: 1 Go Word wrap	K Errors (0)	
<pre>275 /**  * Called when sensor#0 push an event. 277 * Sensor#0 is defined according to what is connected to input pin SensorHandle#0 278 * of this designer block. 278 * @param timestamp 281 * @param data pointer to data event from sensor#0 (orientation) 282 * @param len data size of this event (in bytes) 283 */ 284 static void game_rotation_vector_data_event(uint32_t timestamp, void* data, uint16, 285 { 286 float32_t dataQuaternionQuat[4]; 287 float32_t guat_tmp[4]={0}; 288 float32_t rotationAxisZ; 290 float32_t rotationAxisZ; 291 assert(len == sizeof(VSensorDataQuaternion)); 293 VSensorDataQuaternion* q = (VSensorDataQuaternion*) data; 294 295 // decimate by skipping events 296 if (sw_decimator.cnt &gt;= 0) 297 return; 298 // reset the decimator counter 298 sw_decimator.cnt = sw_decimator.rate; 209 201 202 203 203 204 205 205 205 205 205 205 205 205 205 205</pre>	_t len)	<pre>\$ AuxiliarySensorDesigner.ImplCode - C Editor</pre>
Close	Apply	Close Apply

### **Configure, Build & Debug eMD**

- Gcc/linaro toolchain is preintegrated
- Configure/build/flash are just button clicks
- Eclipse IDE projects generated

	Debug - door/DoorOpeningDetection.inc - Eclipse	- 🗆 ×	
	<u>File Edit Source Refactor Navigate Search Project Run W</u> indow <u>H</u> elp		
	🖉 🎋 Debug 🛛	🍇 🗱 🚺 🗢 🗖 🗍 🎬	3
Image: Second	<pre>vior.elf Thread #1 <main> (Suspended : Breakpoint) vext_listener_game_rotation_vector_callback() at DoorOpeningDetection_wrap.c48 0x2e26 shext_user_thread() at SensorHubExtc:741 0x39ae peningDetection.inc  assert(len == sizeof(VSensorDataQuaternion)); VSensorDataQuaternion* q = (VSensorDataQuaternion*) data; // decimate by skipping events if (sw_decimator.cnt &gt;= 0) return; // reset the decimator counter sw_decimator.cnt = sw_decimator.rate; float angle = 2 *acos(intq30_to_float(q-&gt;w)) * 180 / 3.14159; </main></pre>	utline 🖾	
CustomSensor #3 Execution of 'mingw32-make' Build successful Firmware configu Debuggable firmware Config Flash Fiash	B SEGGER J-Link Debugging] LinkGDbServerCLexe         g breakpoint @ address 0x00002E26, Size = 2         bytes @ address 0x00002E26 (Data = 0x79114A2E)         course       64 bytes @ address 0x20003980         Reading 64 bytes @ address 0x200039C0		

### Live demos

- Rearrange desktop to showcase your ideas
- Get your next round of funding

🜡 SensorStudio - GSH-custom-sensor-ligh	tsaber	– o ×
File Edit View Organize Tools Run	Window Device Help	Sensor studio
🌡 📒 🚍 C 🗘 🔍 🥄	🞦 🗖 🕅 🔨 🖉 🐰 🗮 🗶 🖻 🕐 📰 csh* 🙅 com4* 🚺 🔊 🕼 😫	
Pool 6 × Fiter 5 Appendixed to the second secon	Here         GenericSensorHub Lightsaber custom sensor         This is a more elaborate sample of dustom sensor usage, using Accelerometer, Gyroscope and Game trattom vector sensors.         If you are not familiar with the dustom sensor concept, check the GenericSensorHub custom sensor sample.         To start the sample, plug your GenericSensorHub evaluation board, build and flash your firmware from the Device toolbar. Don't forget to refresh and select the serial COM port.         Then lay the board flat for 5 seconds to allow the gyroscope to calibrate itself.         Here is the description of the algorithm component:         • Start/Stop: hidd the board vertically (Y pointing down, check the Qube panel to see which direction each axis is pointing) 0.5 sec to turn on the lightsaber, 1.5 sec to turn it off.         • Single detection: signifithm will also detect shocks         This sample uses sound, so make sure you've turned sound ont	Properties         F ×           GSH-custom-sensor-lights aber : project         Invé           Author         Invé           Version         1.0.1           Tible
Genrandot Genrandot Gravby GSH Gyroscope Gyroscopet Lineark.ccel Magnetom Magnetom Magnetom Magnetom Dientation Dientation Dientation Stagetom Stagetom Stagetom Dientation	Emulated version Emulated version Cost GameRotationVector GameRotationVector Came	Help • × New project. Properties: • Author (string): Project author. • Version (string): Project version. • Title.
» Fows ≫ Panels	Hend Embedded version Embedded Lightsaber Custom Sensor Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 Power2 Power2 SelectCol Power2 Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2 SelectCol Power2	Console         ≤         ×           Info::STLINK v2 TAG v25         APIv2 SWM v14 VID         APIv2 SWM

## **Migrating to a Full Prototype**

# sensing the

Simple and fast migration from Algo/SW development --> proto/product development





A - Tested and validated project is integrated onto Nucleo via SensorStudio

B - Create new Application code to include rest of the platform needs i.e . connectivity shield



SW Development Environment for GenericSensorHub APP/HW product development Environment

# Working with your final Product sensing the **FUTURE**

- SensorStudio can work with your final product too
- "Dynamic Protocol"
  - code connect SensorStudio & GenericSensorHub
  - source code is available for you to modify if needed



#### Final Product Architecture

# What's new in 2.2? SW

- Record and replay all sensor and algorithm signals

   Repeatability is key to a converging iterative d
- Add INVN sensor fusion & framework on Cortex M4
- C editor :
  - Syntax highlighting
  - search & replace,
- User experience improvements
  - New Welcome Screen
  - Simplified operations (e.g. Device auto connect)
  - Tutorial
- Packaging include eMD, docs
- Still support FireFly<sup>tm</sup> Dev Kit (ICM-30670)

# What's new in 2.2? SW highlights

# sensing the **FUTURE**



#### C Editor



#### Welcome Screen



#### **Record & Replay**



#### Tutorial

# What's new in 2.2? ICM-20690 DK

- Add support for ICM-20690 Development Kit
  - Nucleo-F411RE: Cortex M4 MCU
    - more processing power, FPU, more Flash, more RAM !
    - On board ST-Link debug (~\$300 cost saving for you!)
  - ICM-20690 6-axis (gyroscope & accelerometer)
    - small 2.5x3x0.91 mm (14-pin LGA) package



### **IoT Tech Hackathon**

# sensing the **FUTURE**







InvenSense invites you to create /recognize/score your best Pokémon Go "curved ball" 3D gesture!

The best Pokémon "curved ball" 3D gesture will win a Traxx drone



The IoT Hackathon is designed to allow you to push the boundaries of innovation, reality and technology to create solutions and concepts that can change the world – all in just two days!

Teams of technologists, developers, makers, designers and visionaries will collaborate to explore the next generation of connected devices and solutions in a relaxed environment that encourages you to innovate and create.

Form teams, explore ideas and build prototypes for a panel of industry expert judges all with the help of knowledgeable mentors who will be on hand during the two day challenge. Judges will be looking for innovation, creativity and something with that little bit extra – now is your time to show them something they have never seen before! The IoT Hackathon will not only provide you with the opportunity to win exciting prizes but gain early access to new development platforms, hardware and earn industry recognition from some of the leading IoT product and service provides in the marketplace.

Limited places available!



	пас	kath	on	
Name				
First		Lust		2
Email				-
Phone				
Job Title				
Company Nam	e			
State				
0 of 2 max characte Additional not	n ts			
Captcha				
AVEN	10/	control		
Type the tex	vecy & herris	Cart	MA	
Submit				

Apply for the IoT

InvenSense Developers Conference 2016

# What's new in 2.2? ICM-30670 DK

- Still support FireFly<sup>tm</sup> ICM-30670 Development Kit
  - Still support Arduino Zero
  - ICM-30670 Added capabilities over ICM-30630:
    - 1kHz ODR
    - Using RAM loading to free up Flash
       ICM-30670



### Where to get SensorStudio?

- Download SensorStudio from Developers Corner
  - <u>https://www.invensense.com/sensorstudio/</u>
- Buy SensorStudio Development Kits today
  - from our Platinum Sponsor: CDI



- Order SensorStudio Development Kits online
  - <u>https://www.invensense.com/products/genericsensorhub-development-kit/</u>
  - <u>https://www.invensense.com/products/motion-tracking/6-axis/firefly-development-kit/</u>

- We listen & bring you most valuable sensing features
- Simple and easy to use sensor development tool
  - Graphical interface
  - Custom sensor & algorithm support
  - Full debug and visualization
- Support complete sensor hub on Cortex M4
  - Power and code size optimized
  - Integrated Sensor Framework (GenericSensorHub)
  - Rich feature set with API command access
- Support open hardware platform integration
  - Arduino/Nucleo developers community
  - Fastest path to prototype and full functionality product



sensing the

FUT

# Conclusion



# **Thank You**



# ICM-30670 Develop<sup>t</sup> Environment

# sensing the **FUTURE**

SensorStudio SW



USB

SPI

- <u>SensorStudio</u> Software tool on PC
  Firefly sensor framework loads as a primitive in
- Fireny sensor framework loads as a primitive in SensorStudio.
- Access to framework features and functions
- Custom algo development capabilities in C.
- Debug code with Eclipse
- <u>Arduino Zero</u> Atmel SamD21 microcontroller
  Arduino with sketch "bridge mode" (used as interface card between SensorStudio and Firefly)

v Arduino HW



- Load Final project from SensorStudio into Firefly memory
- Run algo in Firefly and SensorStudio simultaneously -> iterative development
- Debug code in Eclipse C/C++

# **Migrating to a Full Prototype**

# sensing the **FUTURE**

Simple and fast migration from Algo/SW development --> proto/product development



SW Development L Environment for Firefly SOC

A - Tested and validated project is integrated onto Firefly via SensorStudio

B - Transfer "Arduino+Firefly shield" HW from SensorStudio to Arduino IDE environment

C - Create new sketch to include rest of the platform needs i.e . connectivity shield





HW product development Environment

### FireFly™ ICM-30670 Architecture

- Hierarchical Tri-Core Processing
  - ARM M0: Open platform
  - DMP4: Algorithm Accelerator incl FFT Android L features offload
- Integrated Accel & Gyro
- I2C Port for additional sensors
  - le mag, pressure, proximity
- Firefly SOC is a slave sensor hub device, running GenericSensorHub
  - needs to boot form external host MCU
- Efficient Combo: DMP4+CortexM0
  - Higher performance-power ratio than Cortex M4 including FPU
- Available/Free resources\*
  - Memory: ~32KB SRAM (incl FIFO) + ~32KB Flash
  - Compute: 64MHz ARM Cortex M0 + DMP4 Algorithm Acceleration



### **GenericSensorHub in FireFlytm**

- ✓ Robust sensor framework
- ✓ Rapid implementation
- ✓ Optimized for low-power
- ✓ Extensible for any sensor
- ✓ Host AP / MCU agnostic
- Firefly SOC is managed through a FIFO protocol (no registers)
- ✓ Input (command) FIFO for communication No race conditions, Scalable Interface.



AP/MCU

AP/MCU

ICM-30670

FireFly tm

sensing the **FUTURE** 

28

# Working with your final Product FUTURE

- SensorStudio can work with your final product too
- "Bridge" code connect SensorStudio & FireFly<sup>tm</sup> ICM-30670
- "Bridge" source code is available for you to integrate
  - Download <u>ICM30670-eMD\_ArduinoZero</u>
  - ICM30670-eMD\_ArduinoZero\src\Bridge



**Final Product Architecture**