

Toutilo Sound Feedback System

November 2018

Ver. 0.1

Project Aim

To design and implement a sound feedback system for Toutilo.

Project Purposes

The purposes of introducing **sound** as an interaction medium are to reach:

1. Enhance Operation Safety
2. Improve Productivity

Feedback Types

The designed sound feedback can be categorized into two classes:

1. Operation Confirmation Sound
2. Notification / Warning Sound

Operation Confirmation

“User-Driven Feedback”

Audio confirmation will be provided for certain user inputs

- for the purpose of acknowledging the successful reception of inputs
- usually occurs right after a user input is received

Operation Confirmation

HMI Code Number	Transition Description
1	Machine switch on
2	Machine switch off
3	Remote controller connected
4	Remote controller disconnected
5	Motor power connected
6	Motor power disconnected
7	Usage scenario activated

Notification / Warning Sound

“Machine-driven feedback”

Notification / Warning will be generated actively based on a machine's perception or its internal states

1. Normal event notification
2. Emergency Warning

Notification / Warning Sound

HMI Code Number	Transition Description
1	“Idle Mode” Enter
2	“Idle mode” to “working mode”
3	“idle mode” to “maneuver mode”
4	“idle mode” to “autonomous mode”
5	“working mode” to “idle mode”
6	“maneuver mode” to “idle mode”
7	“autonomous mode” to “idle mode”
8	“working mode” to “autonomous mode”

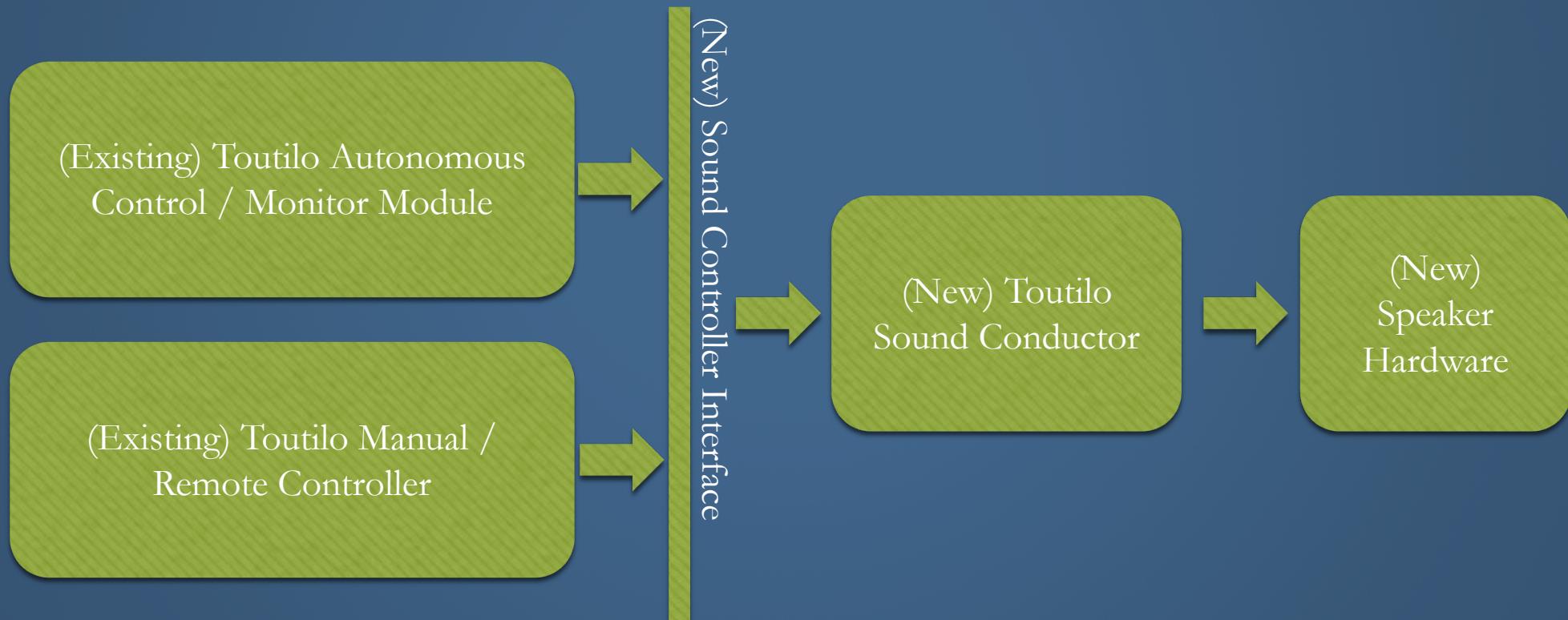
Notification / Warning Sound

HMI Code Number	Transition Description
9	Idle mode duration exceeds xx hours
10	Abnormal Battery State
11	Abnormal Left Motor State
12	Abnormal Right Motor State
13	Abnormal Leveling Motor State
14	Abnormal Tool Motor State

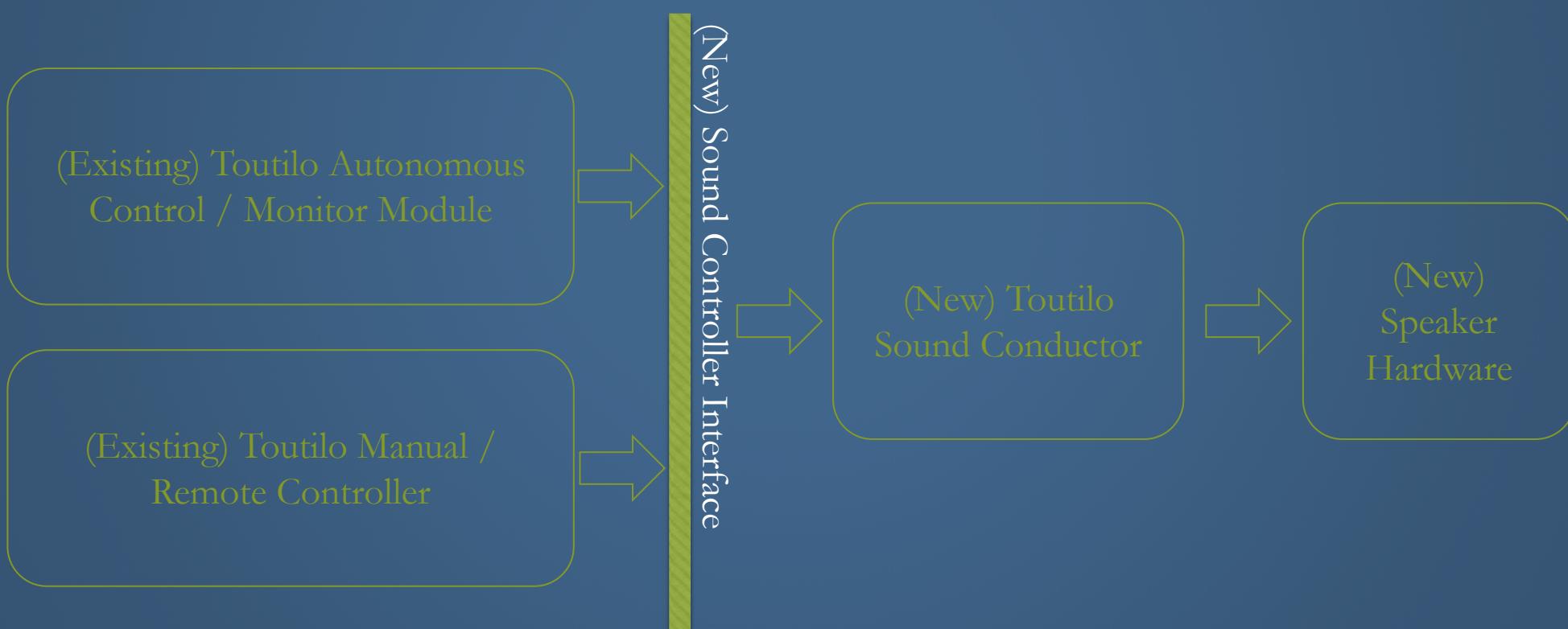
Notification / Warning Sound

HMI Code Number	Transition Description
15	Approaching the end of field (planned turning point)
16	Deviating from the planned path
17	Any other emergency events (?)

System Architecture Overview (Ver. 1)



Sound Controller Interface

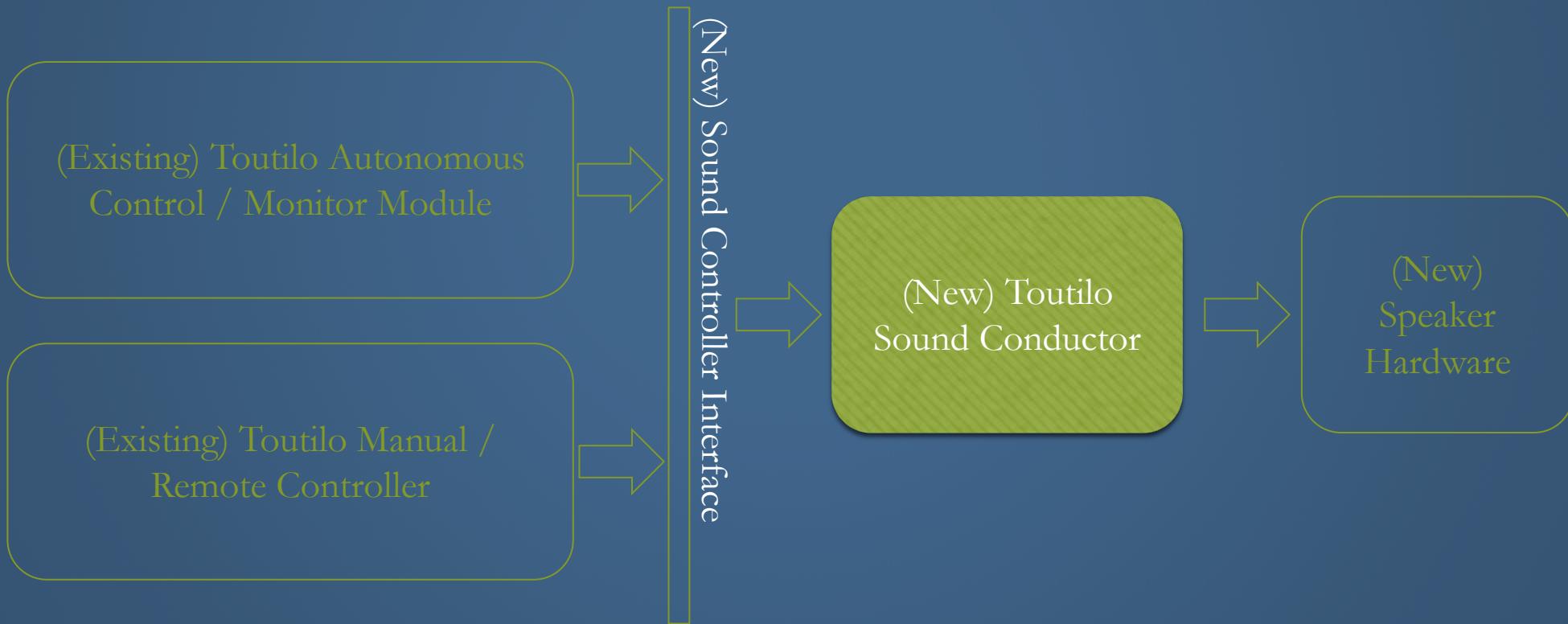


Sound Controller Interface

Build on top of Tutilo's existing inter-vehicle communication protocol

- Using the *publisher-subscriber* communication paradigm (?)
- The interface is defined as a communication channel
- All components that need to trigger any sound event publish to this channel
- Asynchronous communication

System Architecture Overview (Ver. 1)



Toutilo Sound Conductor

(New) Toutilo
Sound Conductor

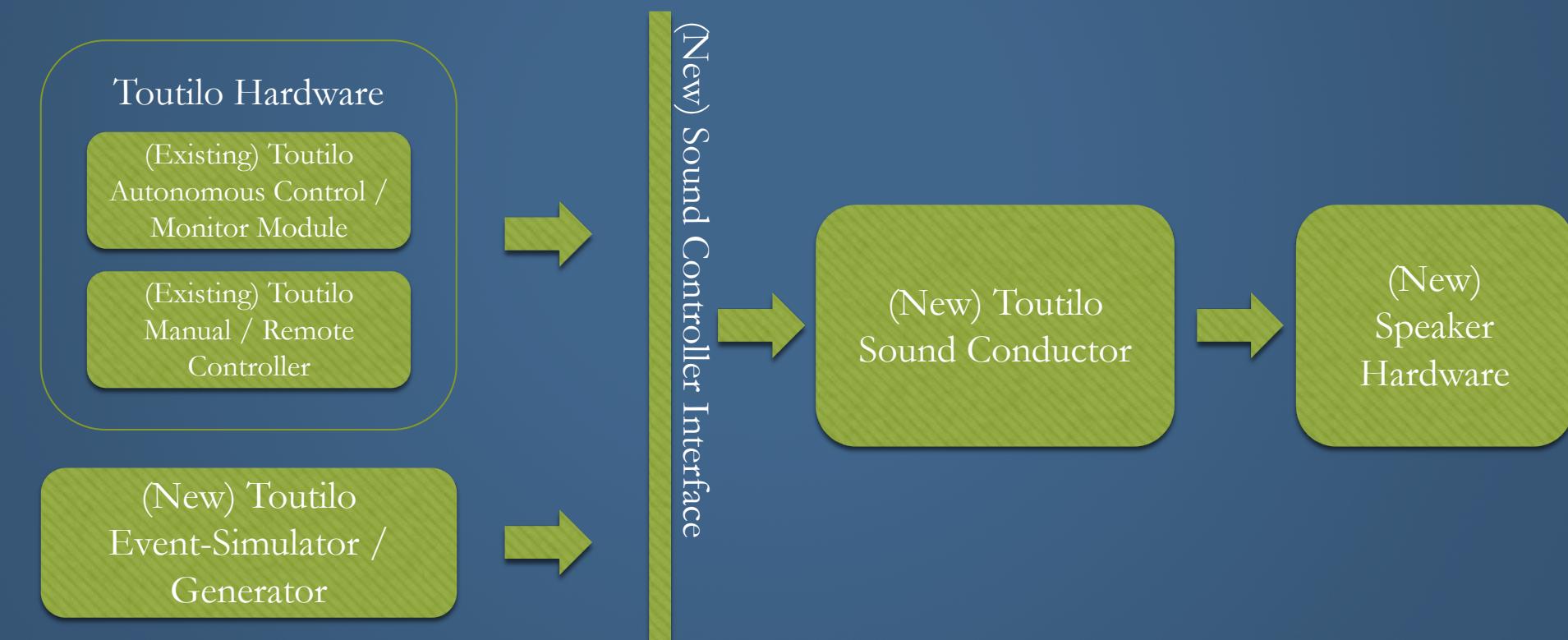
(Engineer) Reconfigurable

- Using an *event-to-sound language* to define the sound effect for any event from the interface
- Define the duration / loudness / level of importance, etc.

(Finite State) Sound Controller

- Coordinated access to the speaker hardware
- To address any necessary sound effect overlay

System Development Model



Toutilo Event-Simulator (Generator)

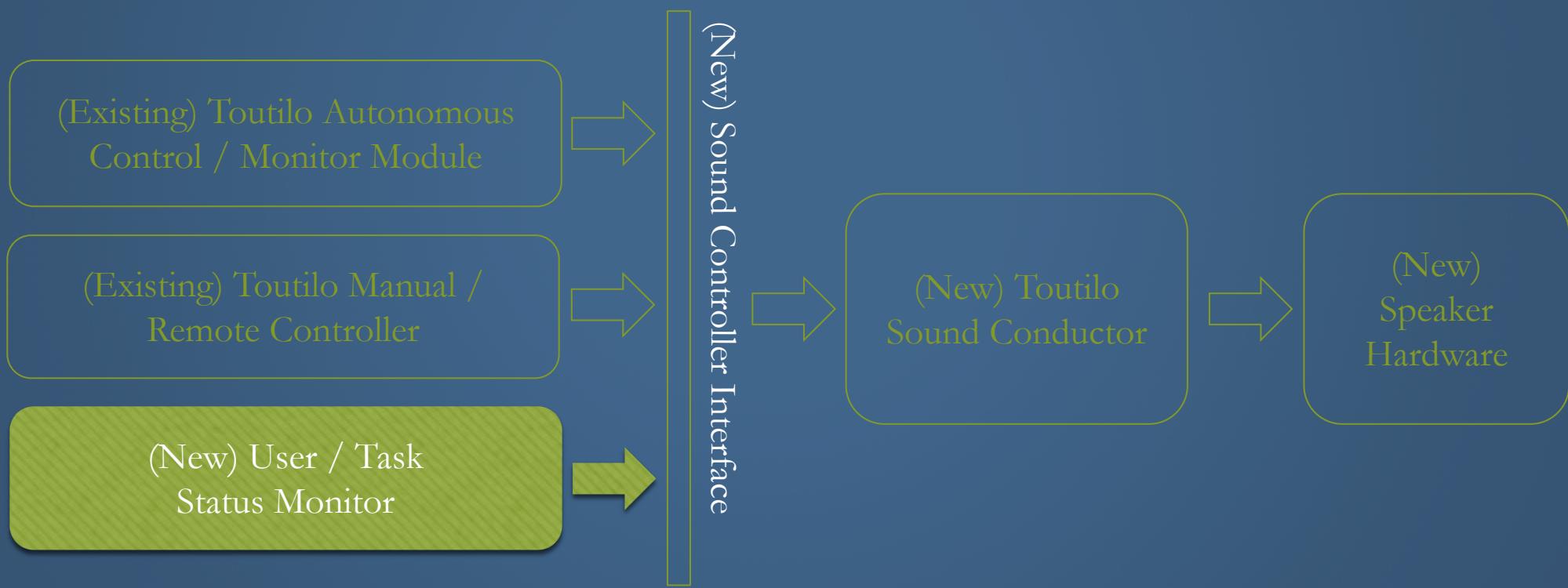
(New) Toutilo
Event-Simulator /
Generator

Share the same communication interface as the real Toutilo hardware

- Using the same publisher-subscriber communication paradigm
- Publish to the same communication channel

Simple user interface to let developers to manually trigger events

System Architecture Overview (Ver. 2)



User / Task Status Monitor

In addition to providing sound feedback based on information collected from the machine, we also want to take “user status” and “task status” into consideration. E.g.

(New) User / Task
Status Monitor

- User Fatigue Monitoring
- *“Sound gamification”*
 - Incomplete task completion notification or
 - positive reward for successful task completion

Extending the Ver 1 design
Requiring additional user / task monitoring capabilities (hardware + Software)

Next Step ...

INRIA & Swansea

- Agreeing on the overall system design
- Agreeing on the *Sound Controller Interface*

INRIA

- Providing sample code for accessing machine information using the *Sound Controller Interface*

Swansea

- Developing *Tutilo Sound Conductor*
- Developing *Tutilo Event-Simulator / Generator*