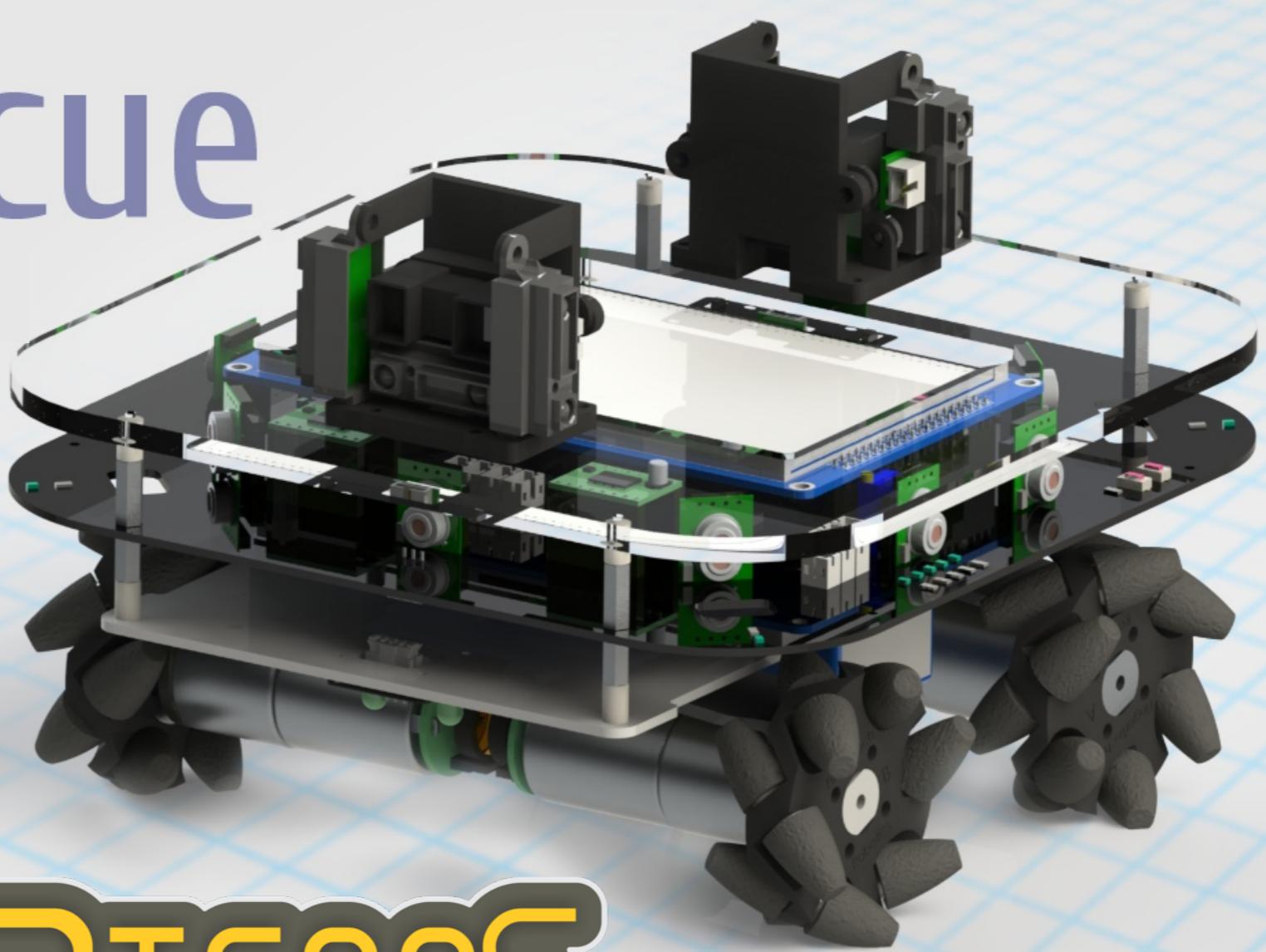


# RESCUE B ROBOT

An Omni-directional robot for the RoboCup Junior

Rescue



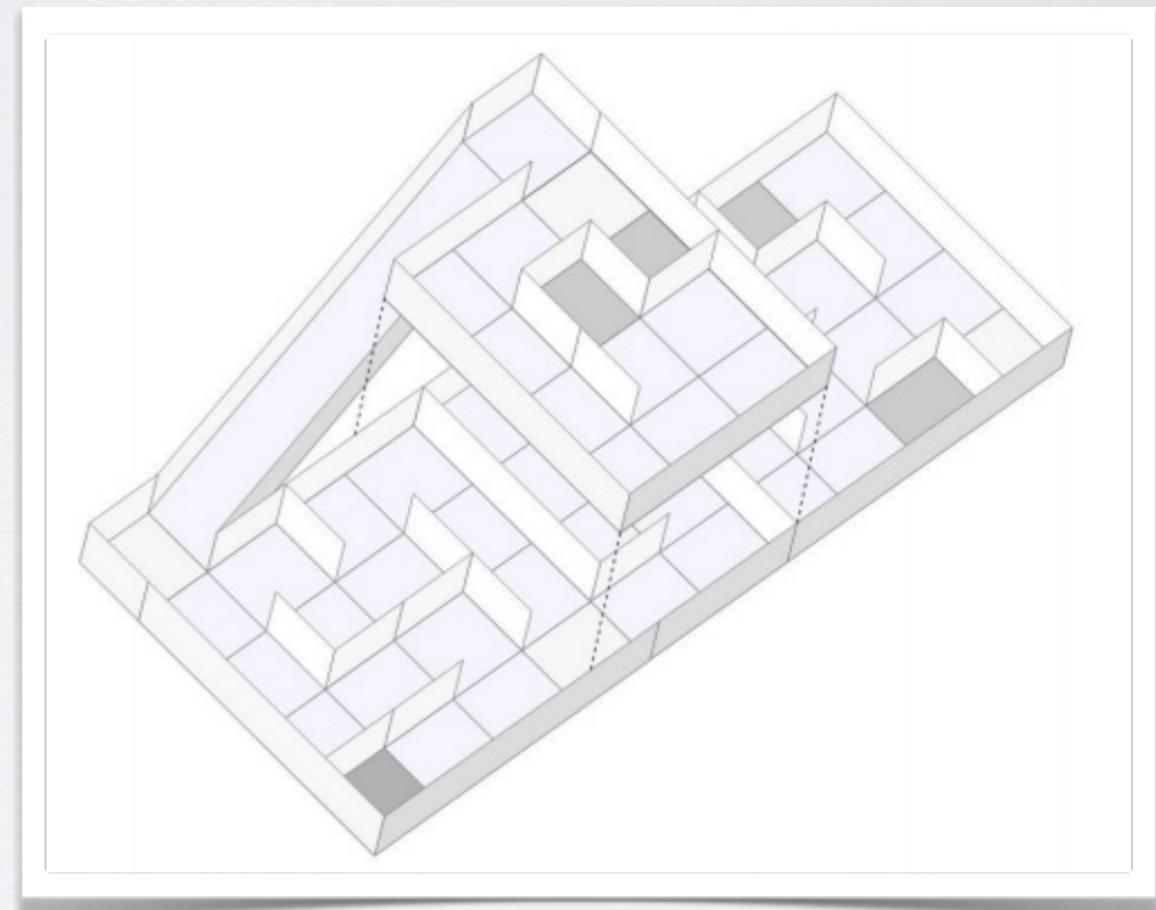
EMER<sup>Ω</sup>TECOS

# GOAL

What is Rescue B, and how does it work

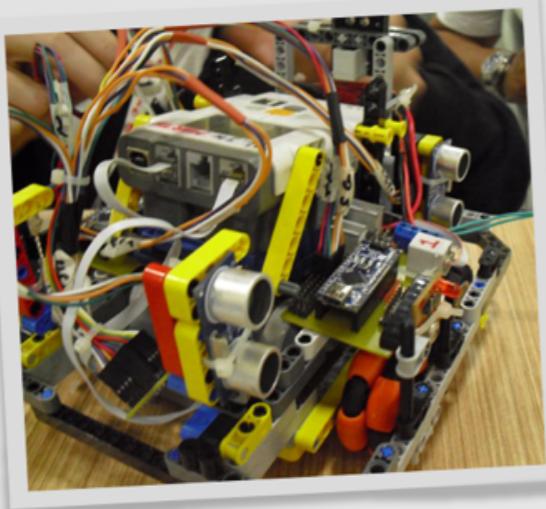
# GOAL

- Find victims
- Come back to first position



# RESCUE.EVOLUTION

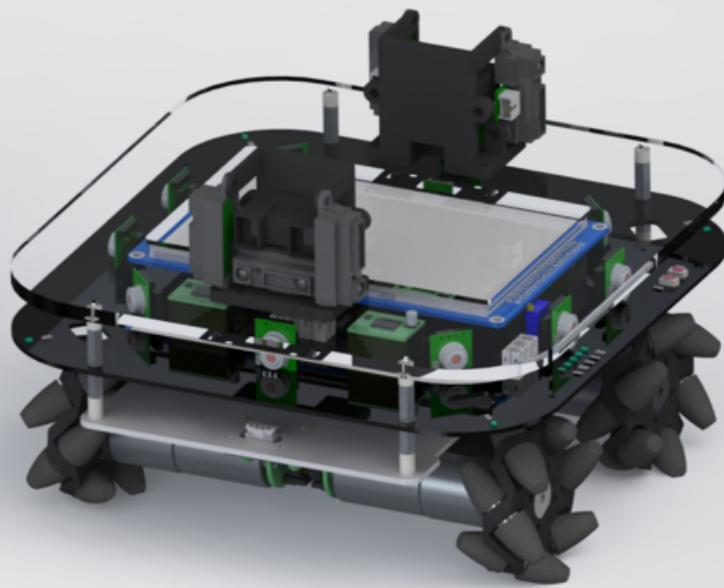
## Rescue B - evolution —



2011



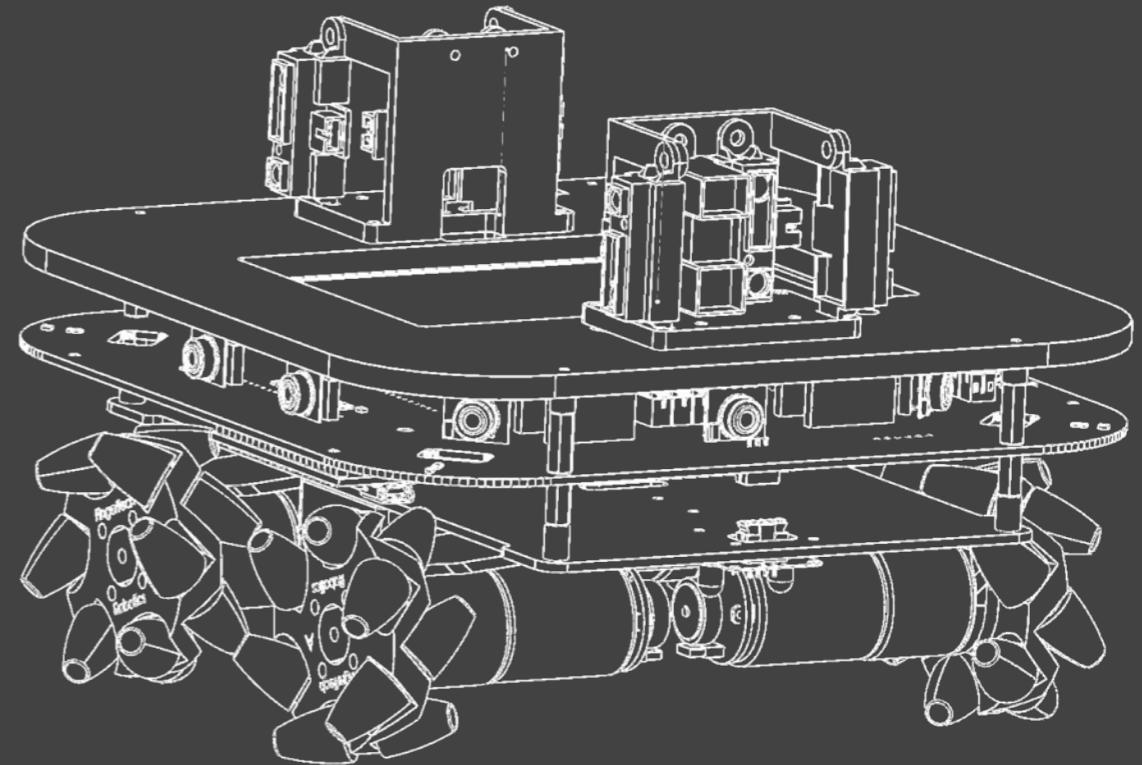
2012



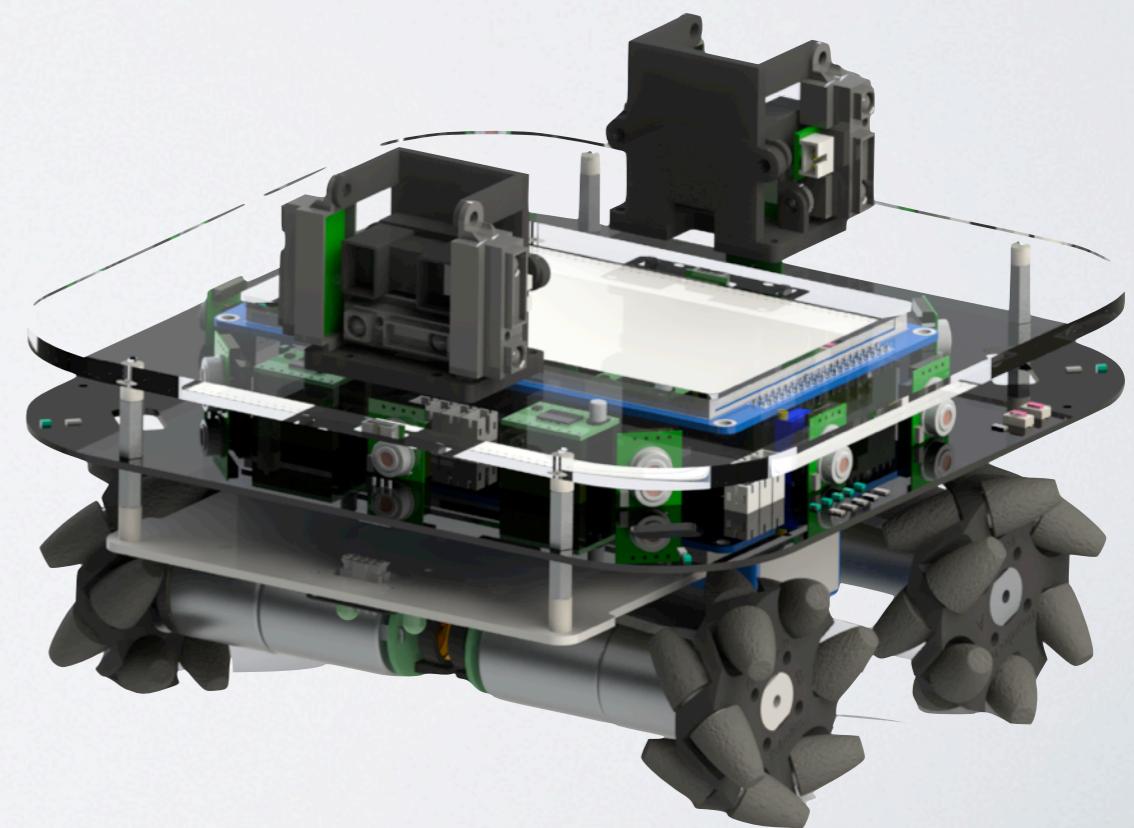
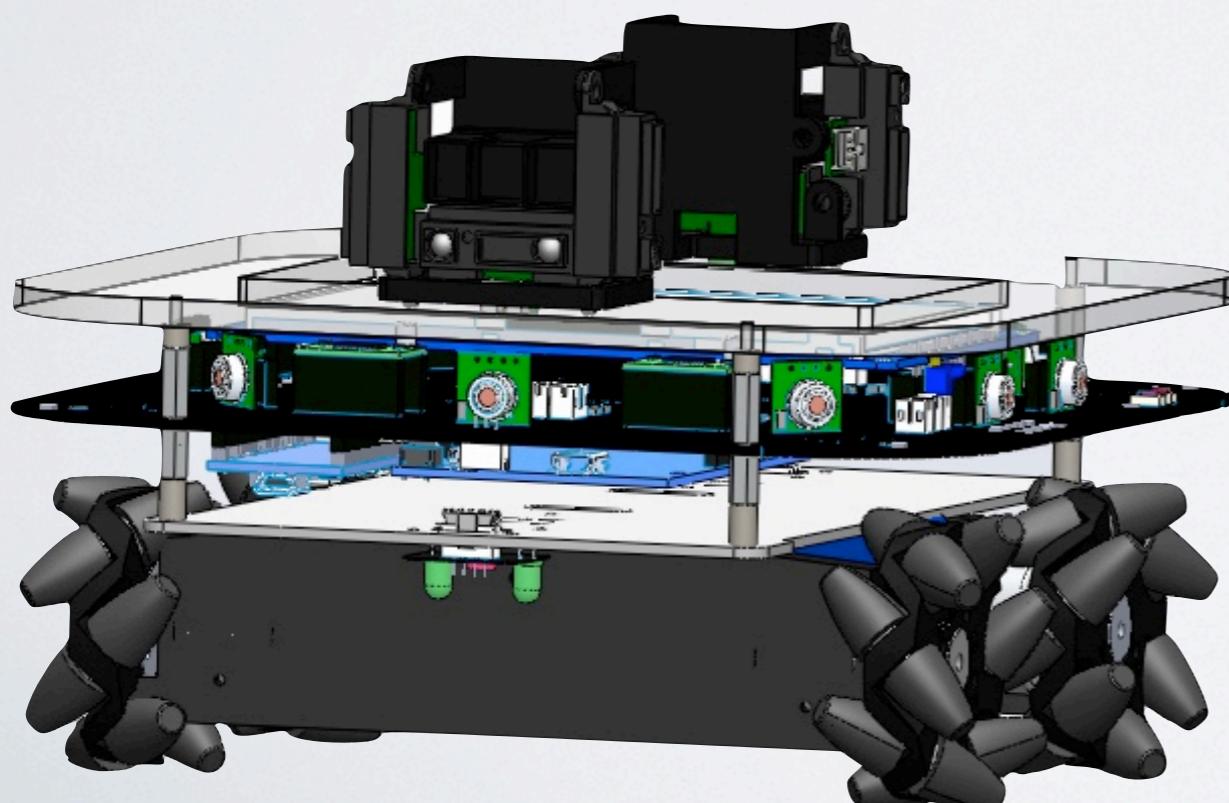
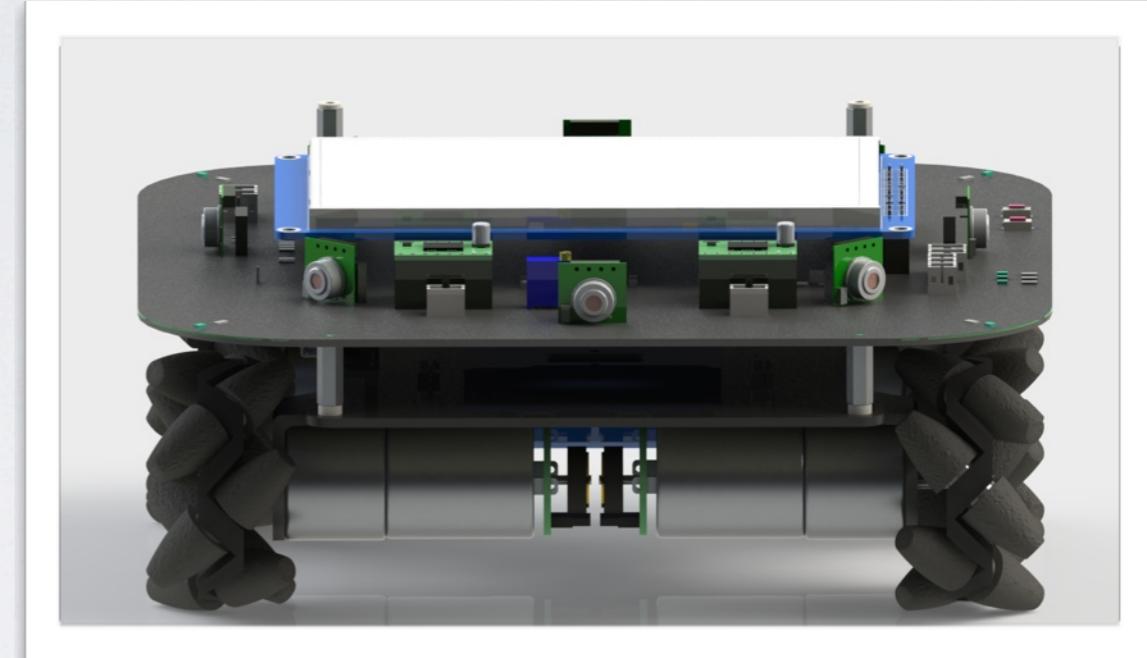
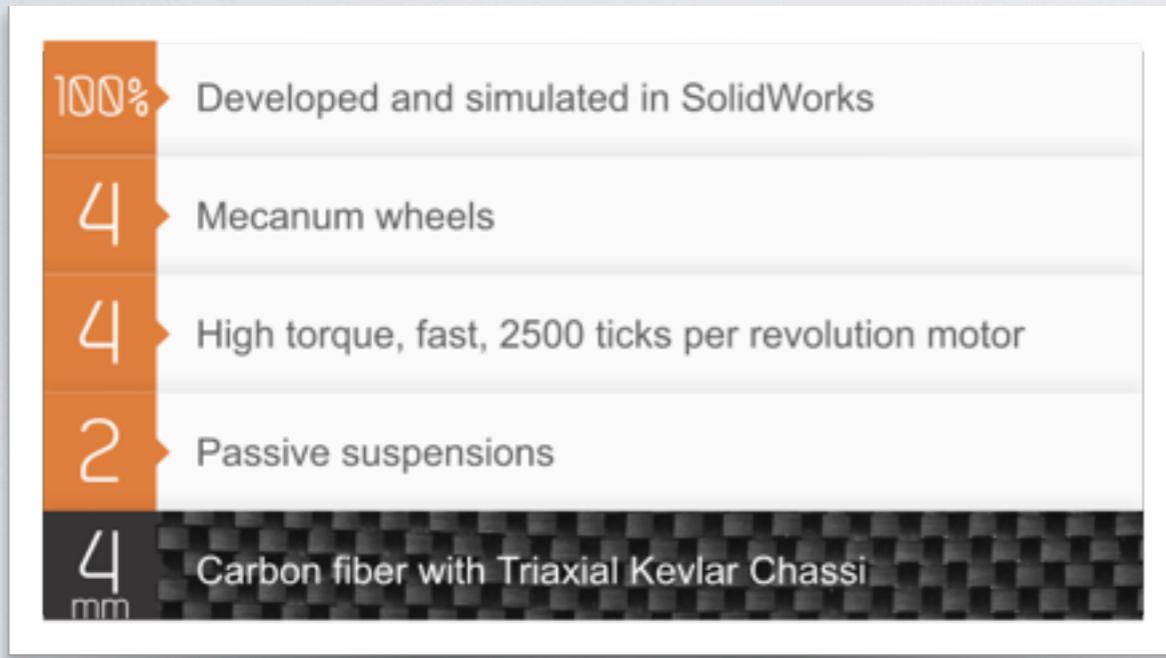
2013

# RESCUE.MECHANICS

Design matters.



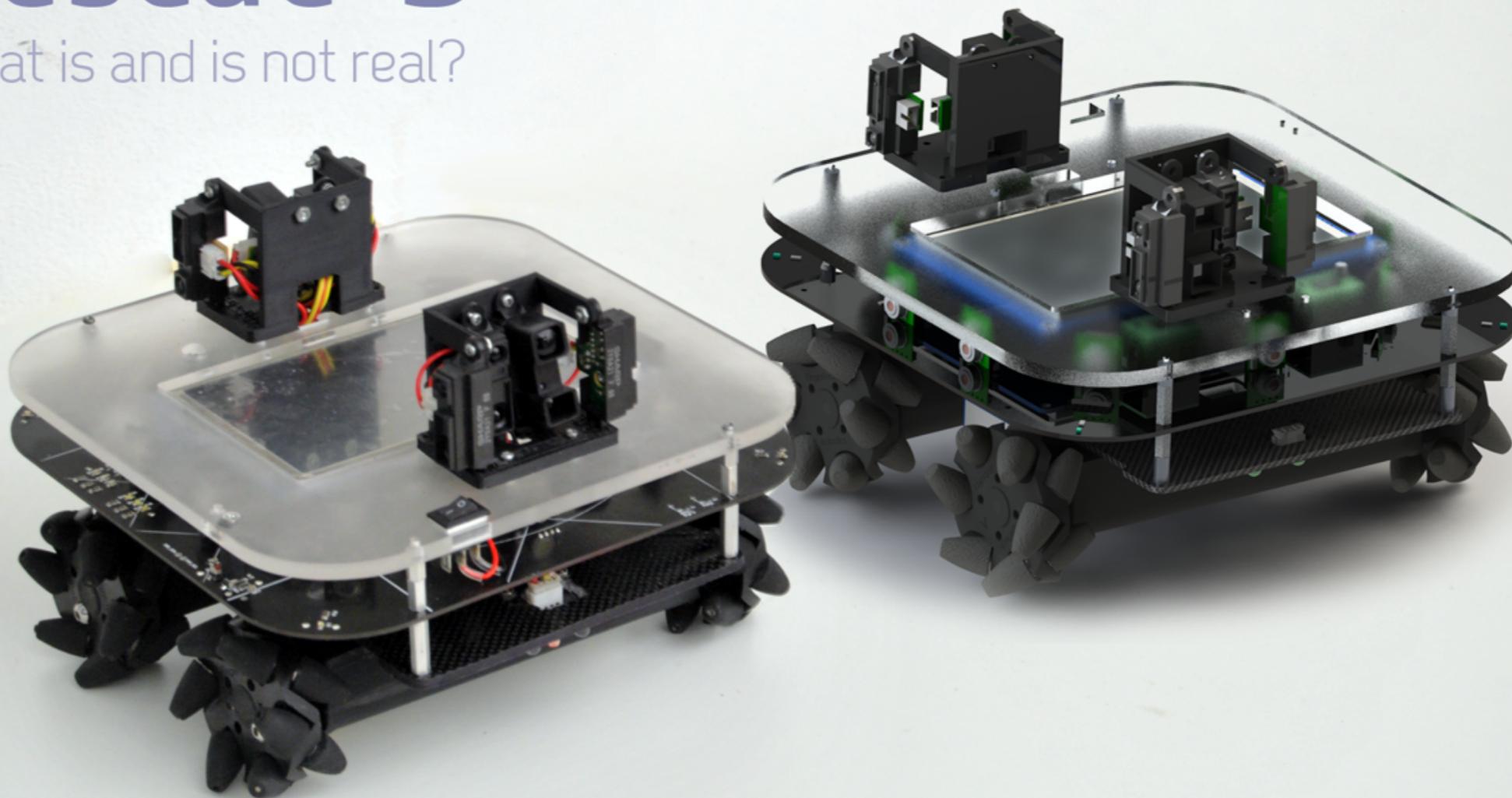
# RESCUE.MECHANICS



# RESCUE.MECHANICS

## Rescue B

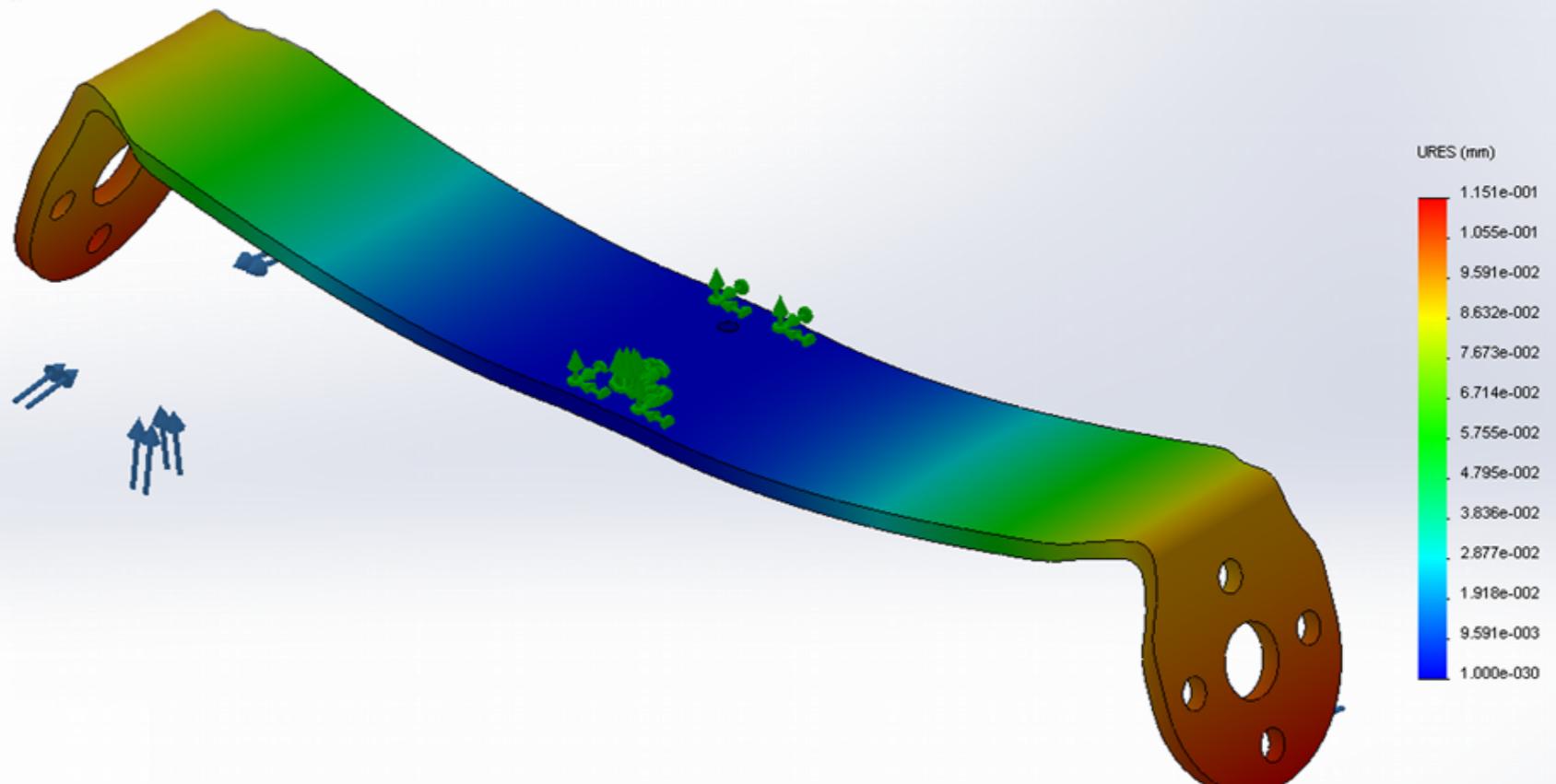
What is and is not real?



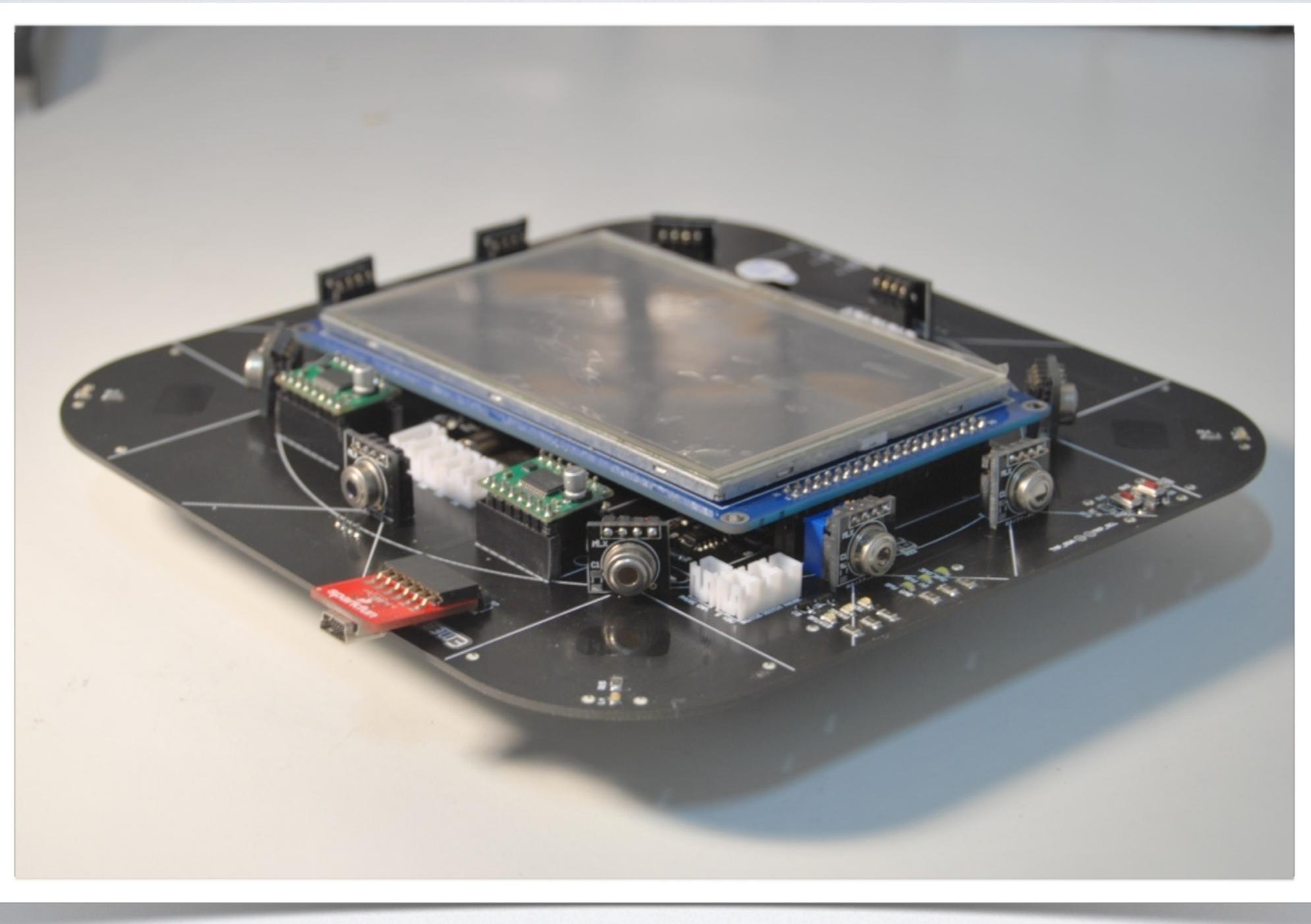
# RESCUE.MECHANICS

## Rescue

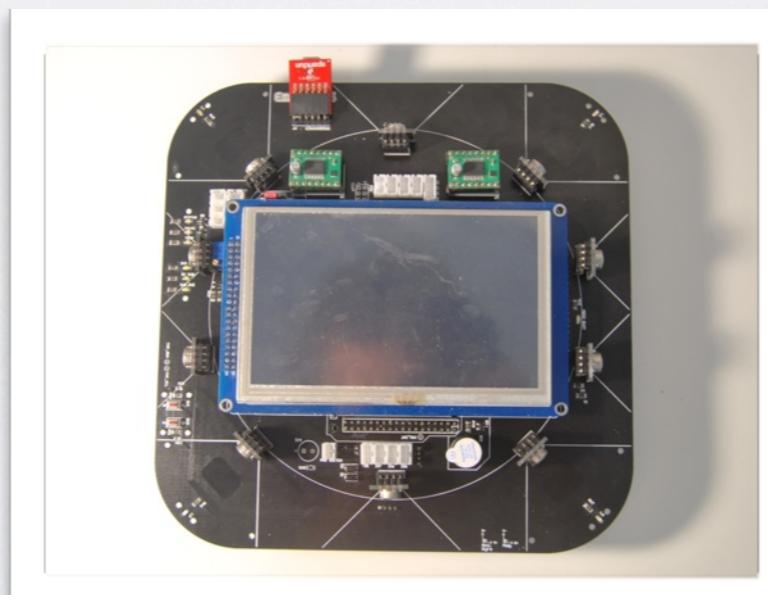
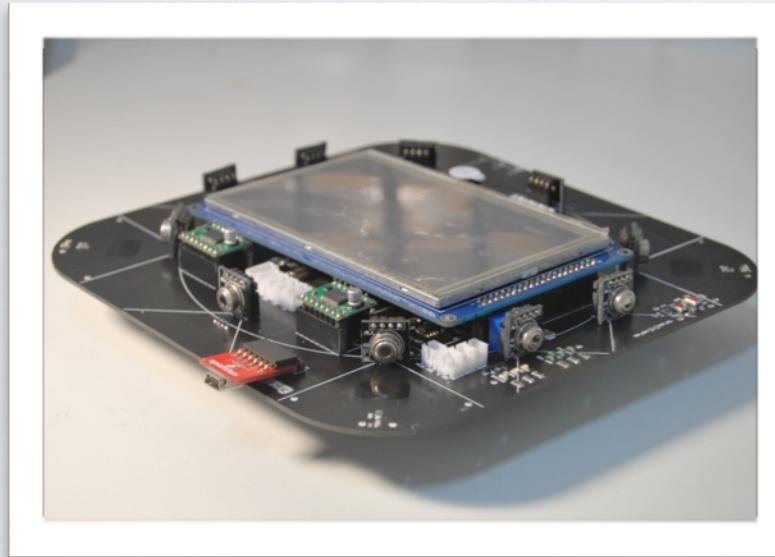
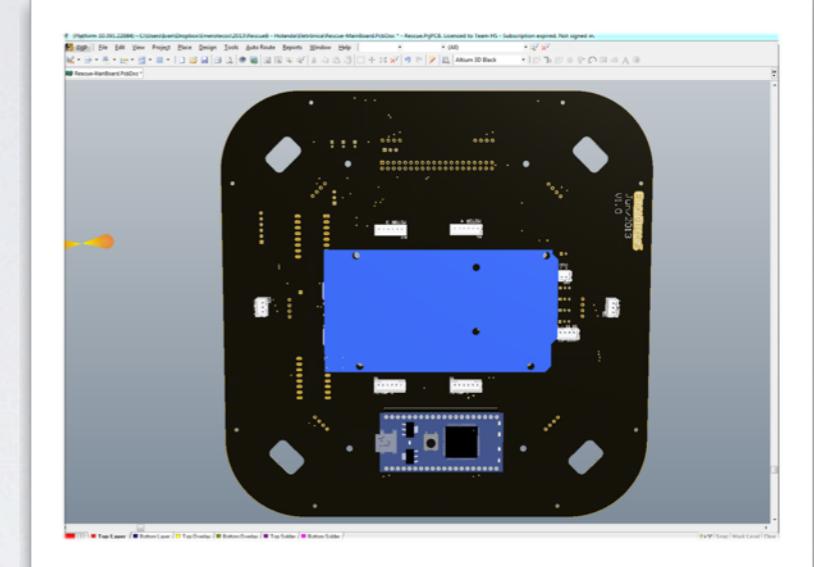
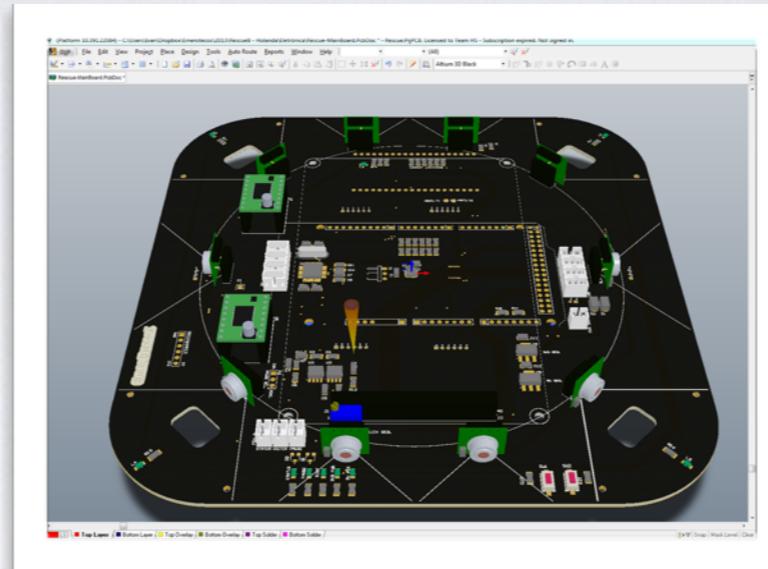
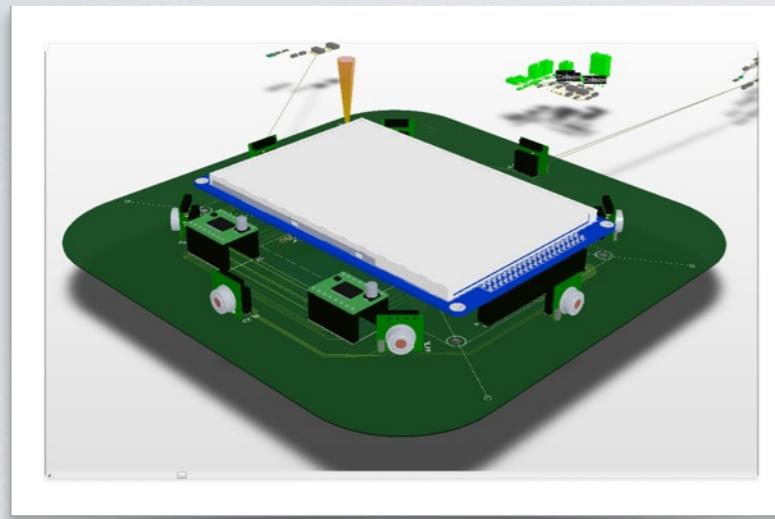
vai aguentar?



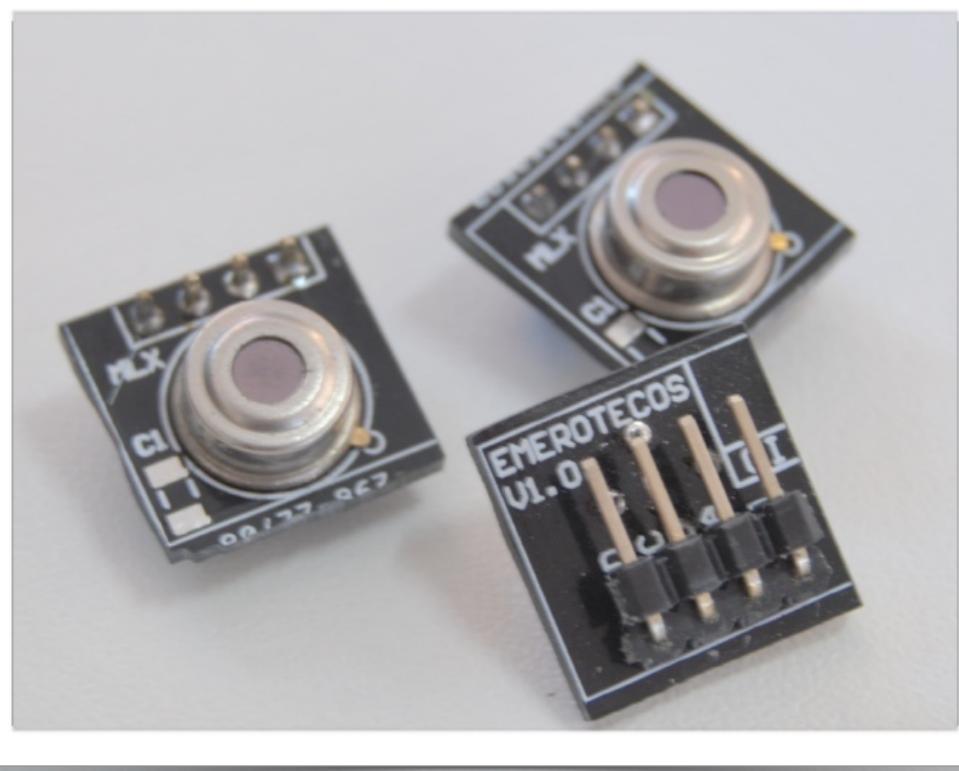
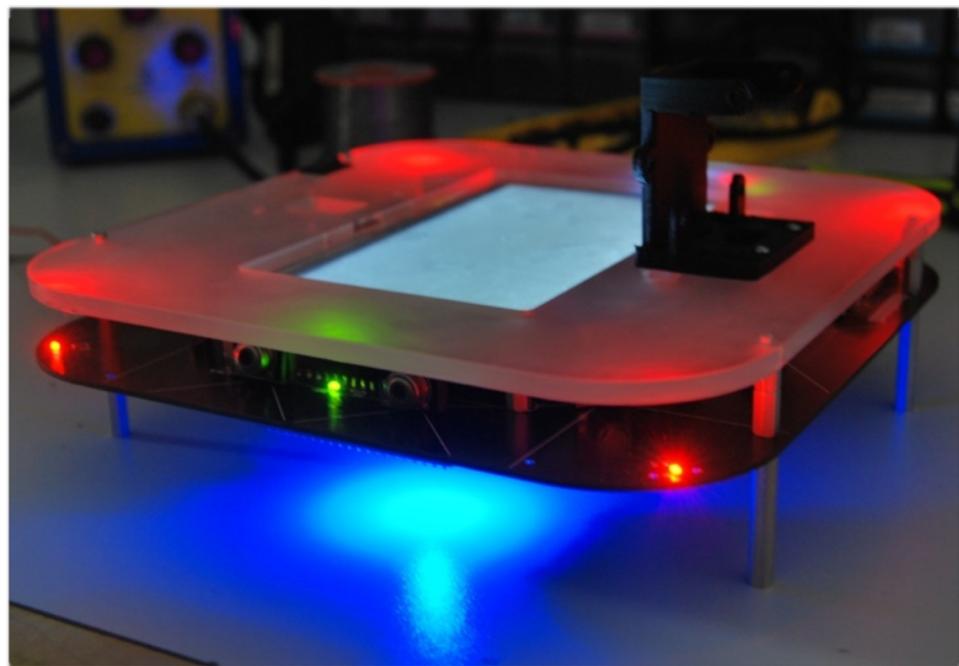
# RESCUE.ELECTRONICS



# RESCUE.ELECTRONICS



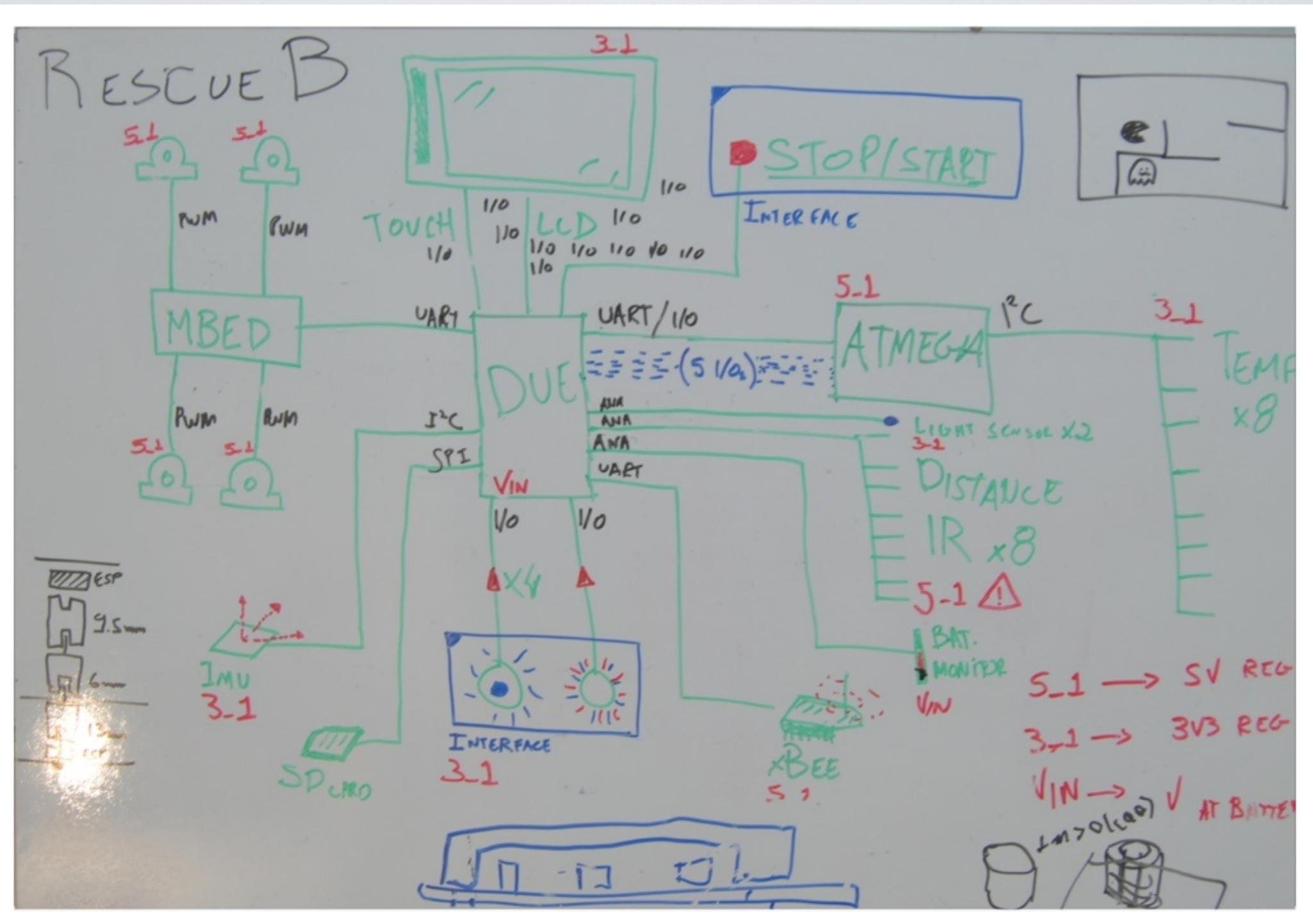
# RESCUE.ELECTRONICS



- 8 Infra-red Distance Sensor
- 10 Infra-red Temperature Sensor
- 2 Active Light sensor
- 1 9 axis IMU with Sensor Fusion, Quaternion output
- 1 5" TFT LCD 800x480 with Touch screen
- 1 Serial Bluetooth
- 2 ARM Cortex M3 - 100Mhz
- 1 Atmega328 - 16Mhz



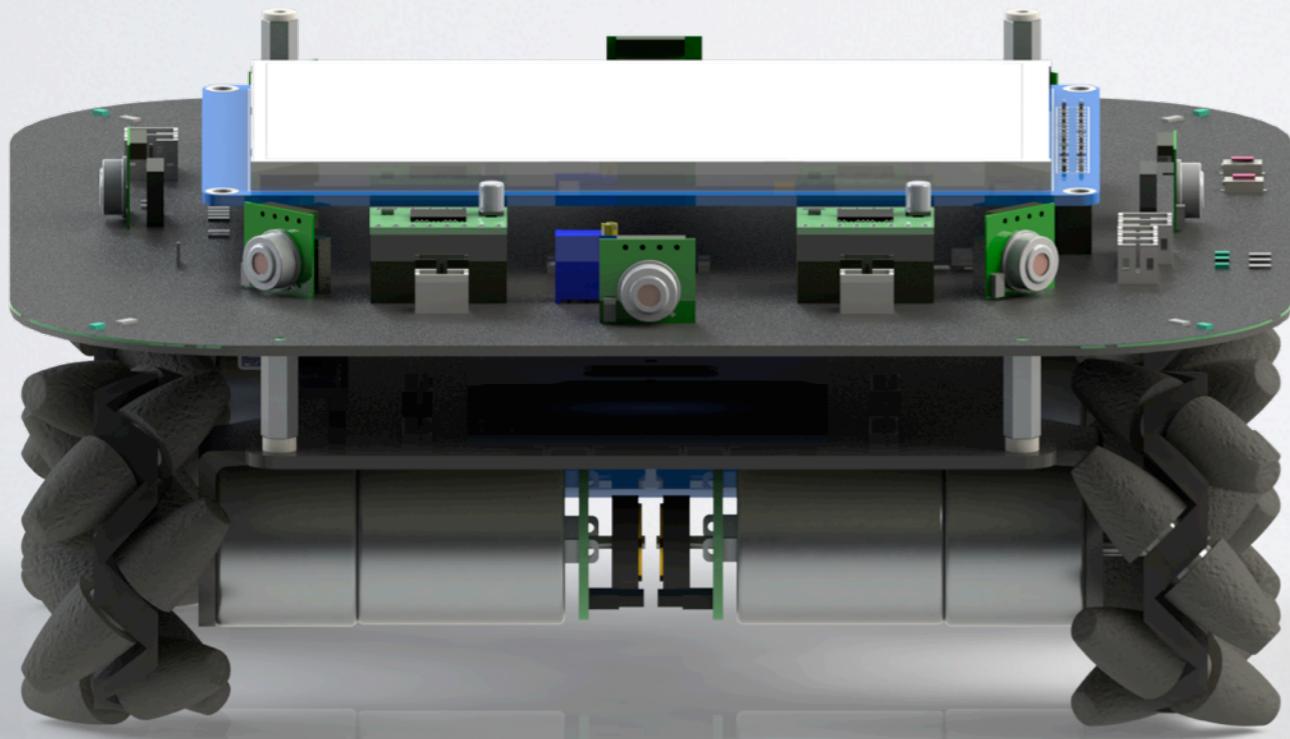
# RESCUE.ELECTRONICS



Schematic

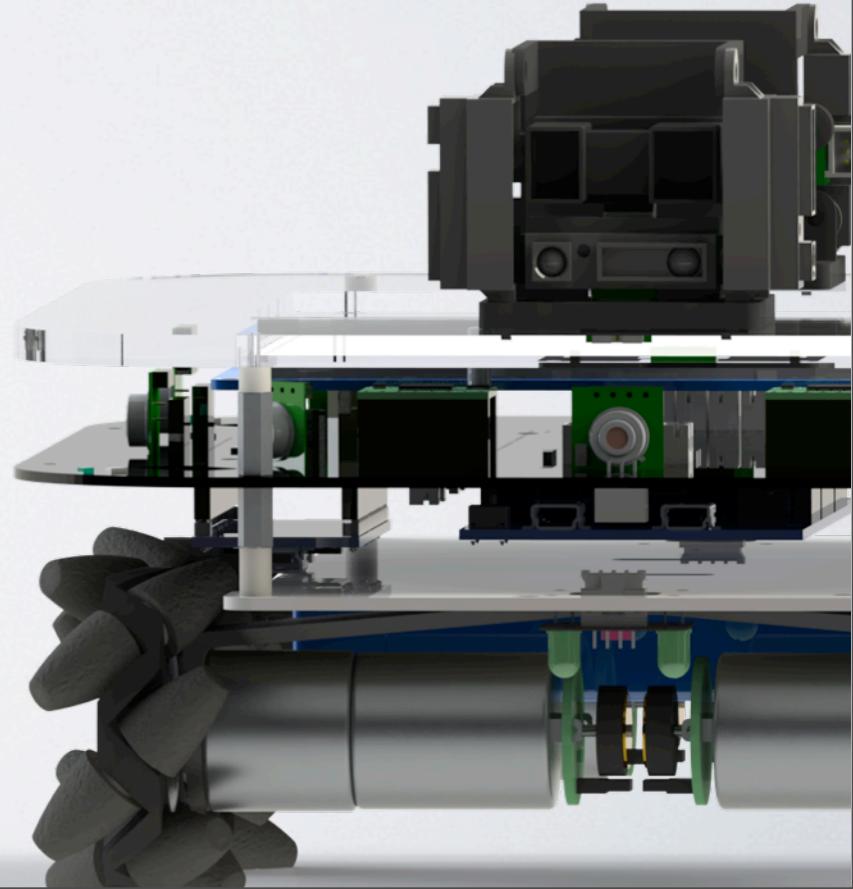
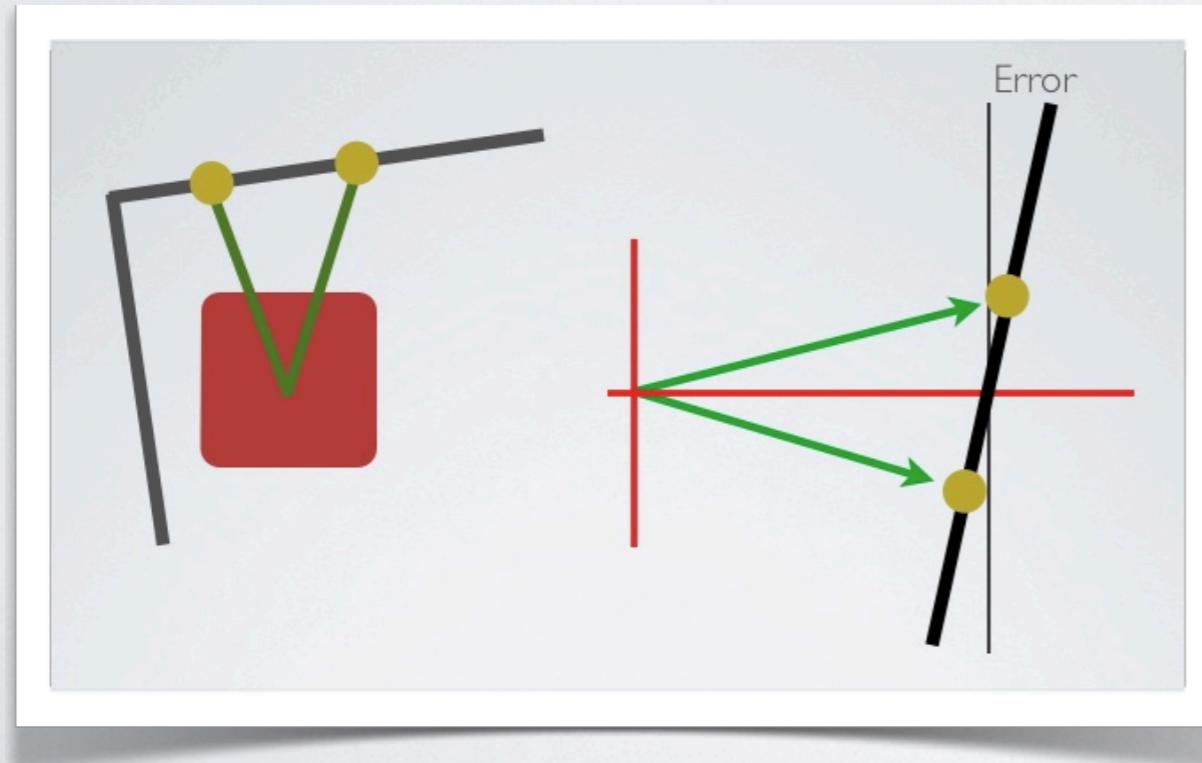
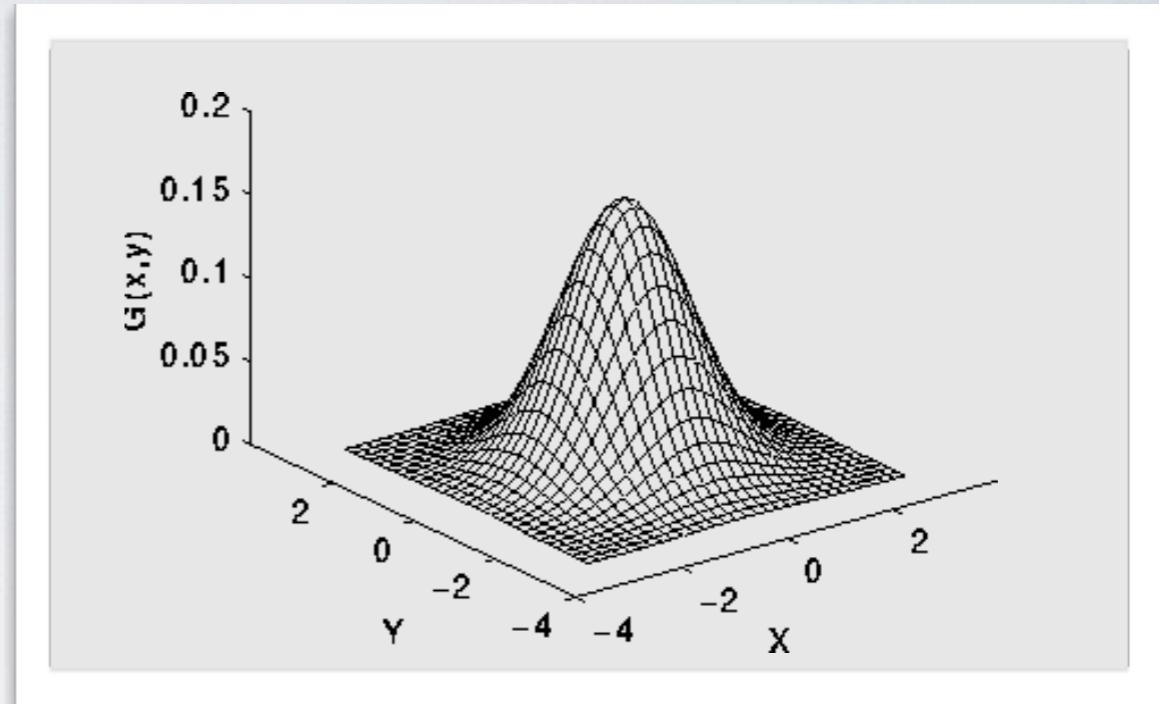
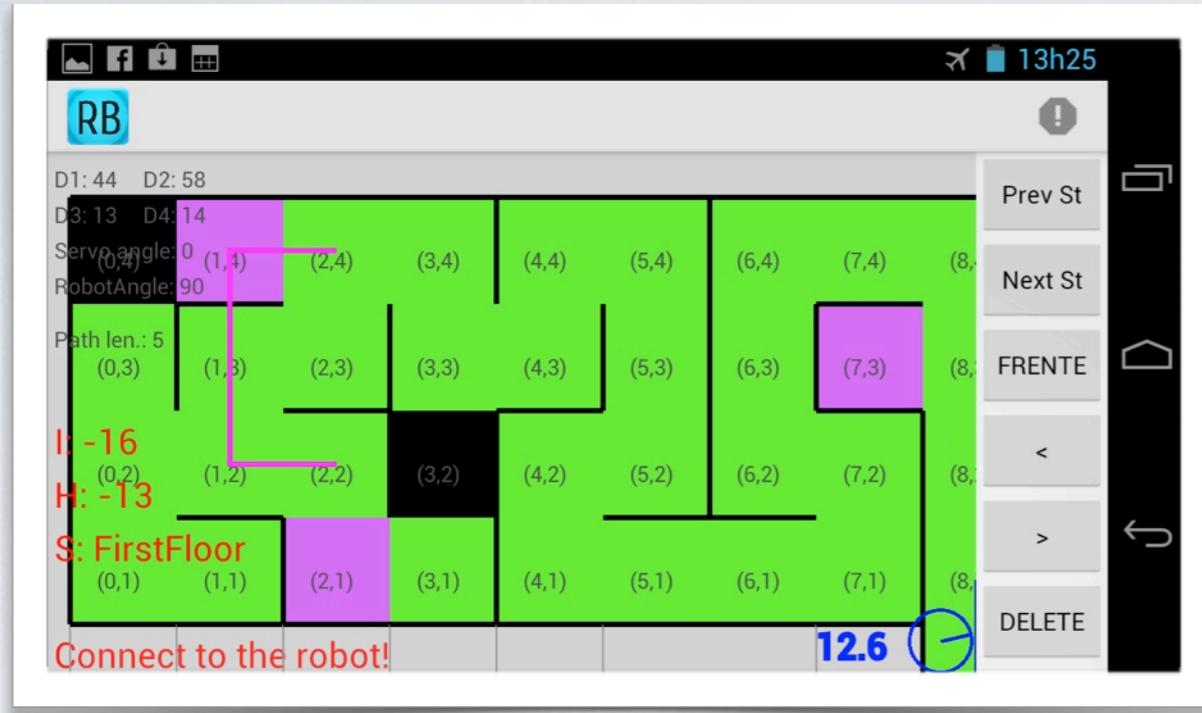
# RESCUE.SOFTWARE

- Custom Operational System for ARM Cortex
- Complete Object Oriented software
- 2D Simulator
- Custom, simple particle filter and Kalman Filter
- Custom powerfull heuristic for solving maze
- Vector based Victim detection

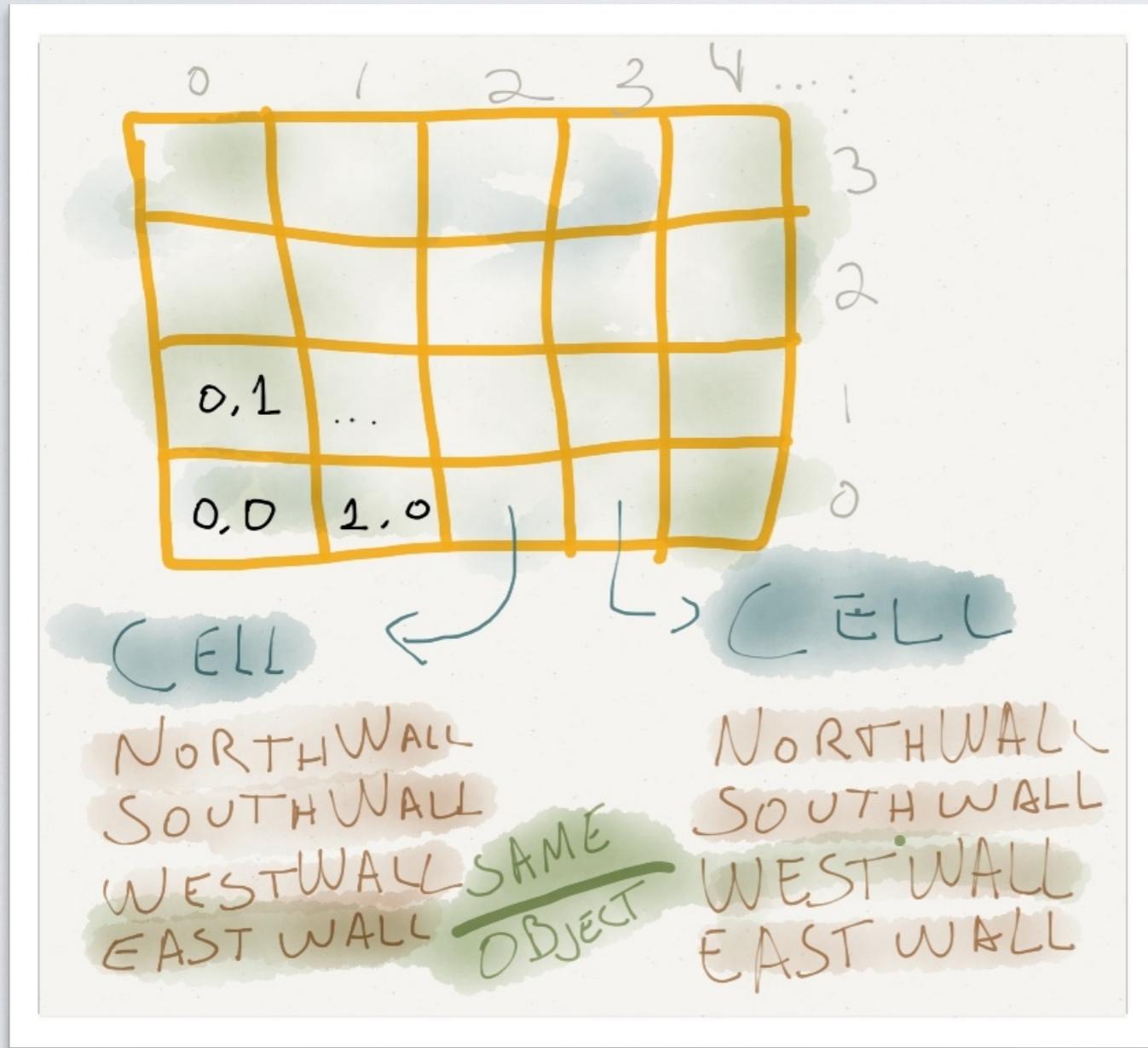


```
256  /*
257   * ArduinOS setup
258   */
259
260  // Thread initialization
261  ArduinOS:: _threadTimer = &TIMER_THREAD1;
262  ArduinOS::SystemThreads = new ThreadController();
263  ArduinOS::SystemThreads->ThreadName = "SystemThread";
264  ArduinOS:: _threadInterval = 10000;
265
266  // Configure Return button of ArduinOS
267  attachInterrupt(pBTN_SW2, irqReturnBtn, RISING);
268
269 #ifndef DISABLE_LCD
270     // Configure ArduinOS to read Touch interrupts from
271     ArduinOS:: _touchInterruptPin = pTCH_IRQ;
272     ArduinOS:: _touchTimer = &TIMER_TCH;
273     ArduinOS:: _touchObject = TCH;
274     ArduinOS:: _touchMode = ArduinOS::INTERRUPT_TIMER;
275     ArduinOS:: _touchTimerPeriod = 70000;
276     ArduinOS::LCD = &LCD;
277
278     // Configure GUI
279     GUI.onRender(onGUIrender);
280     ArduinOS::GUI = &GUI;
281     STAGE.onRender(onSTAGEmode);
282     ArduinOS::STAGE = &STAGE;
283
284     // Create and register GUI updater on SystemThreads
285     thrUpdateGUI = new Thread(onGUIupdate, UPDATE_GUI_IN);
286     thrUpdateGUI->ThreadName = "Update GUI";
287     ArduinOS::SystemThreads->add(thrUpdateGUI);
288     PC.println("$: GUI Updater Thread registered: "+thrU
289 #endif
290
291     // snippet beep() register on ArduinOS
292     ArduinOS::beep = beep;
293
294     // Beeping initialization
295     thrBeeper = new Thread(onBeeper, TIME_BEEP_H); // Insta
296     thrBeeper->ThreadName = "Beeper";           // Default T
297     thrBeeper->enabled = false;                 // Defalut i
298     ArduinOS::SystemThreads->add(thrBeeper);    // Register
299     PC.println("$: Beeper Thread registered: "+thrBeeper->Th
300
```

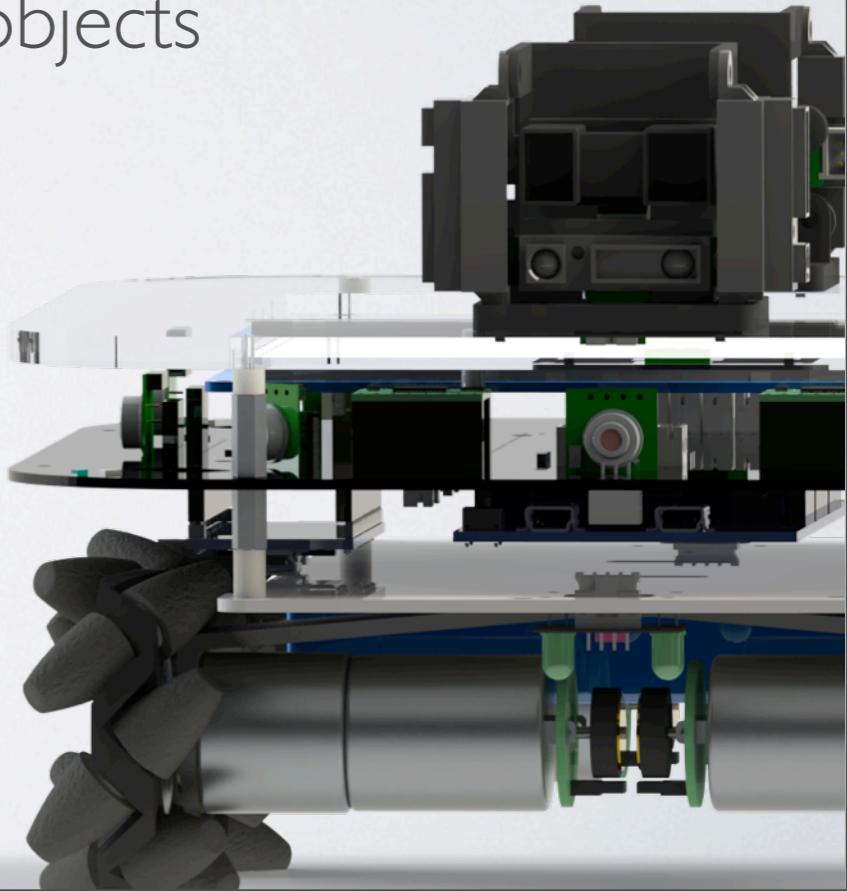
# RESCUE.SOFTWARE



# RESCUE.SOFTWARE

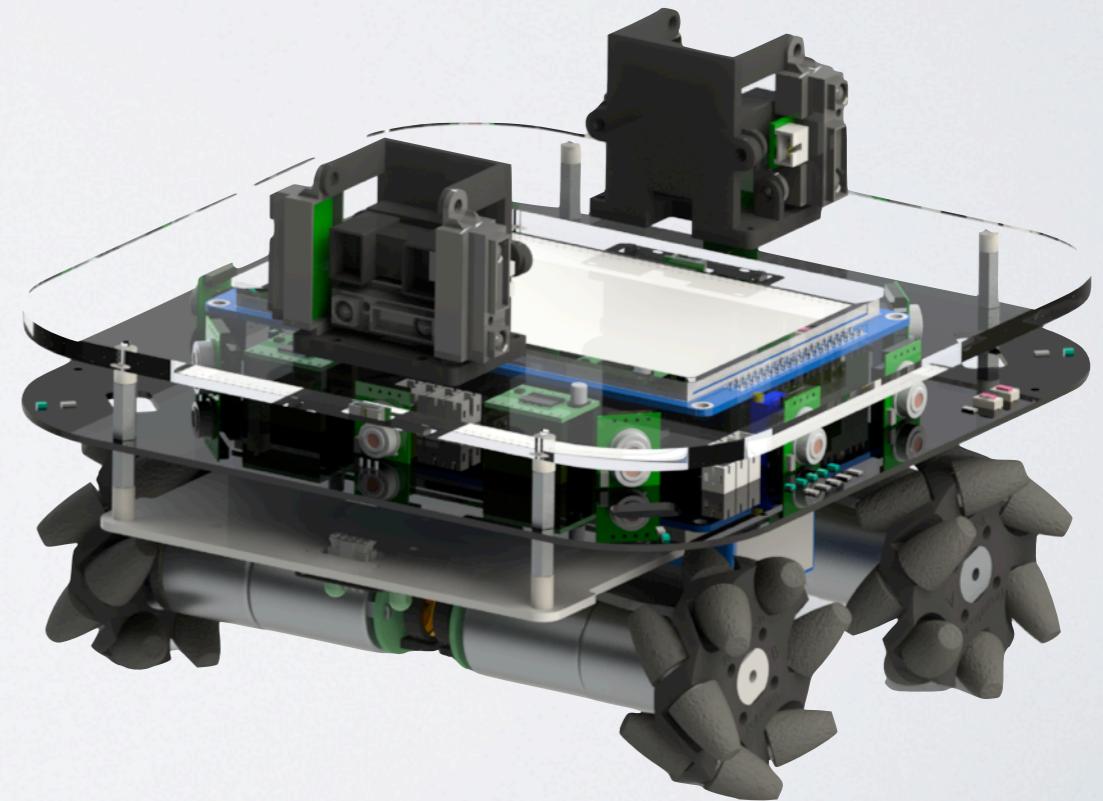
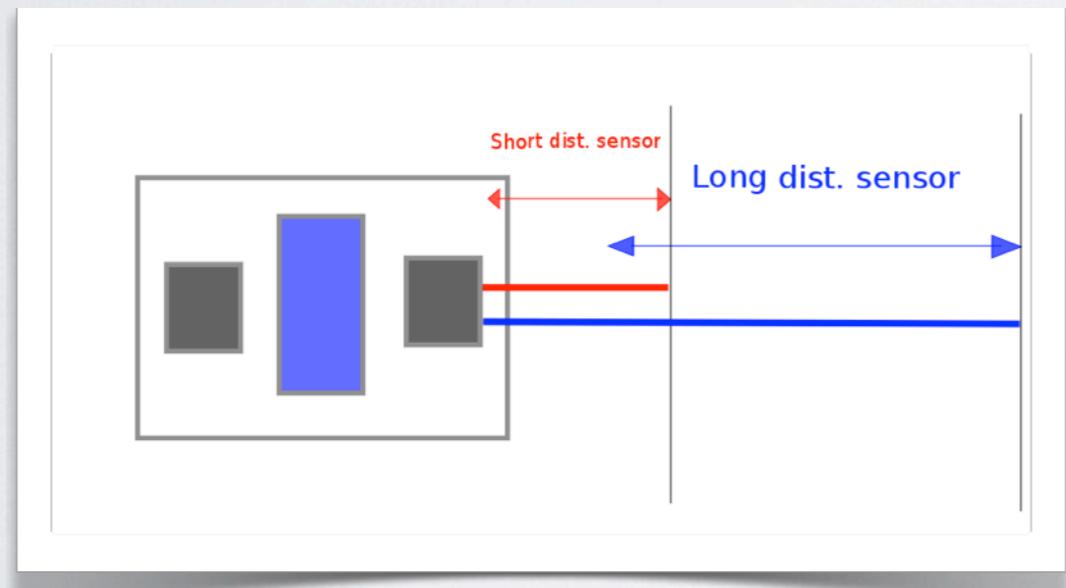


- Probability
- Existence
- Non Existence
- Are objects



# RESCUE.PROBLEMS

- Too much Planning
- Durable and resistant Suspension
- Long and short range sensors
- Infrared “resonance” frequency





Abilio Marcos, Andre Seidel, Ivan Seidel, Matheus Canejo.  
Mentor: Felipe Nascimento

**TendaDigital.net**  
**NossosRobos.blogspot.com**