**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM  38911 MAX BYTES FREE

READY

# RAPID ROBOT PROGRAMMING

PAUL J. PERRONE

Perrone Robotics

# TALK GOALS

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- HOW-TO ON CREATING ROBOTS
- MORE RAPID ROBOT PROGRAMMING
- USING JAVA + HIGHER LEVEL LIBRARIES
- SENSING, ACTUATION, & PLANNING

# ROBOT SOFTWARE IN A NUTSHELL

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

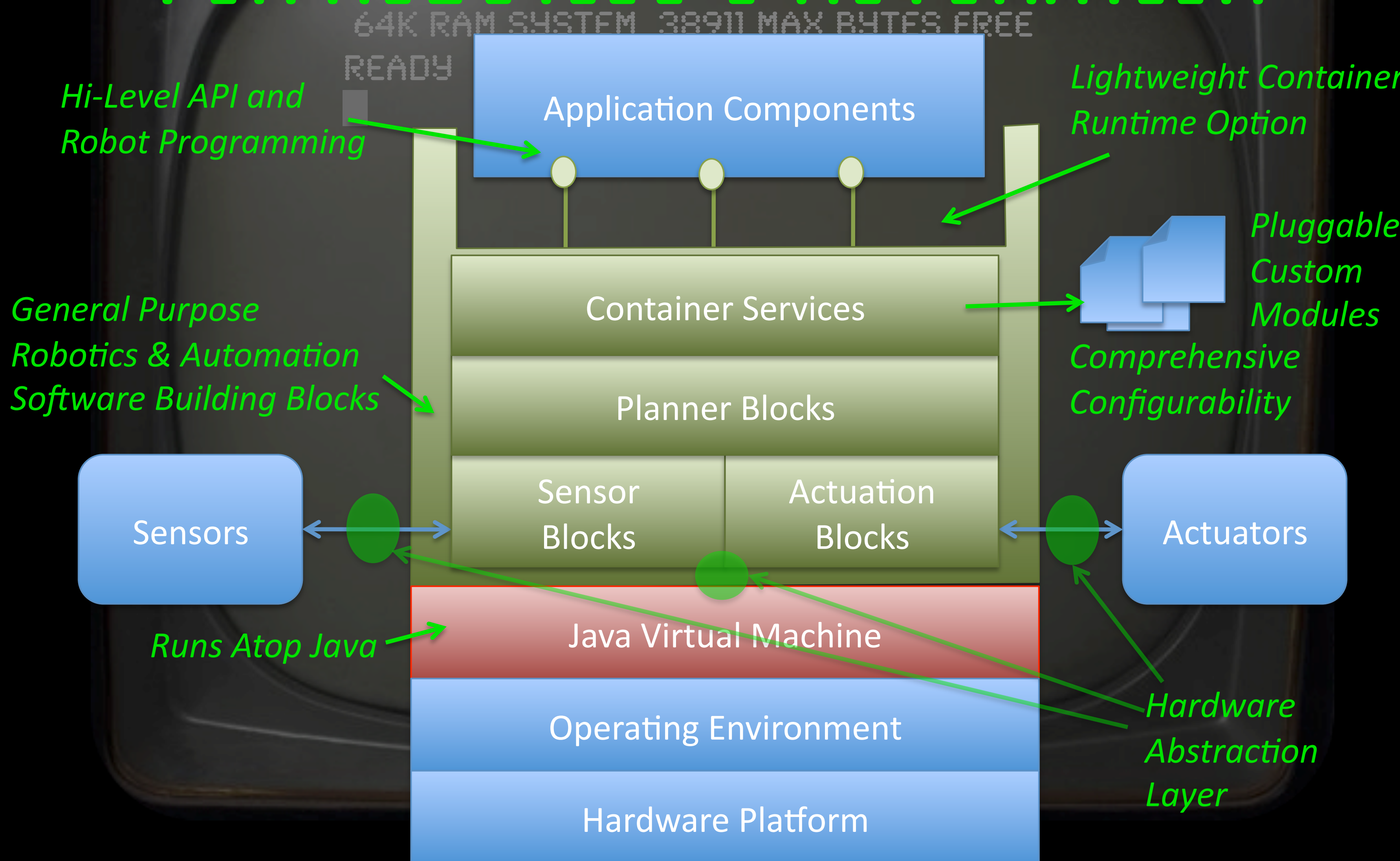
READY

## SENSE...

## PLAN...

## ACT...

# MAX: FLEXIBLE BUILDING BLOCKS FOR ROBOTICS & AUTOMATION



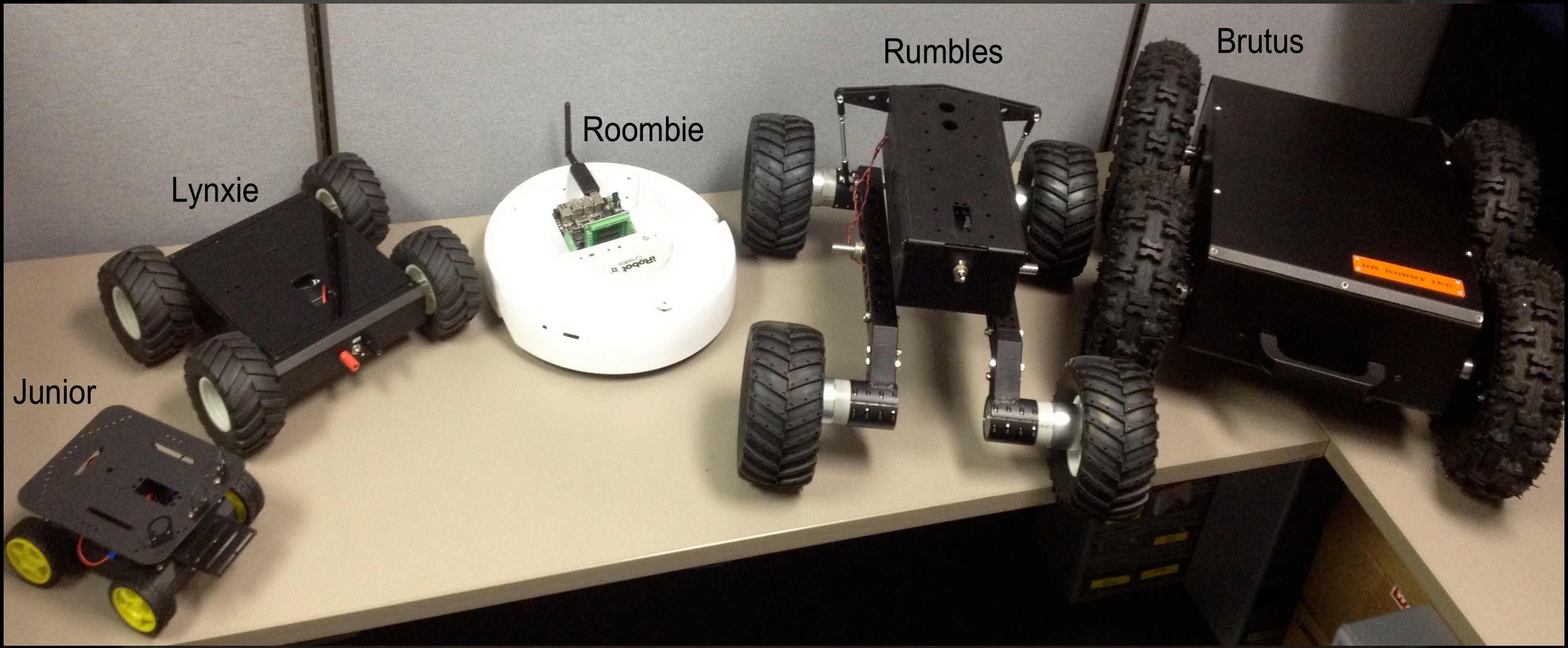
# THE AARDBOTS: A ROBOT FAMILY

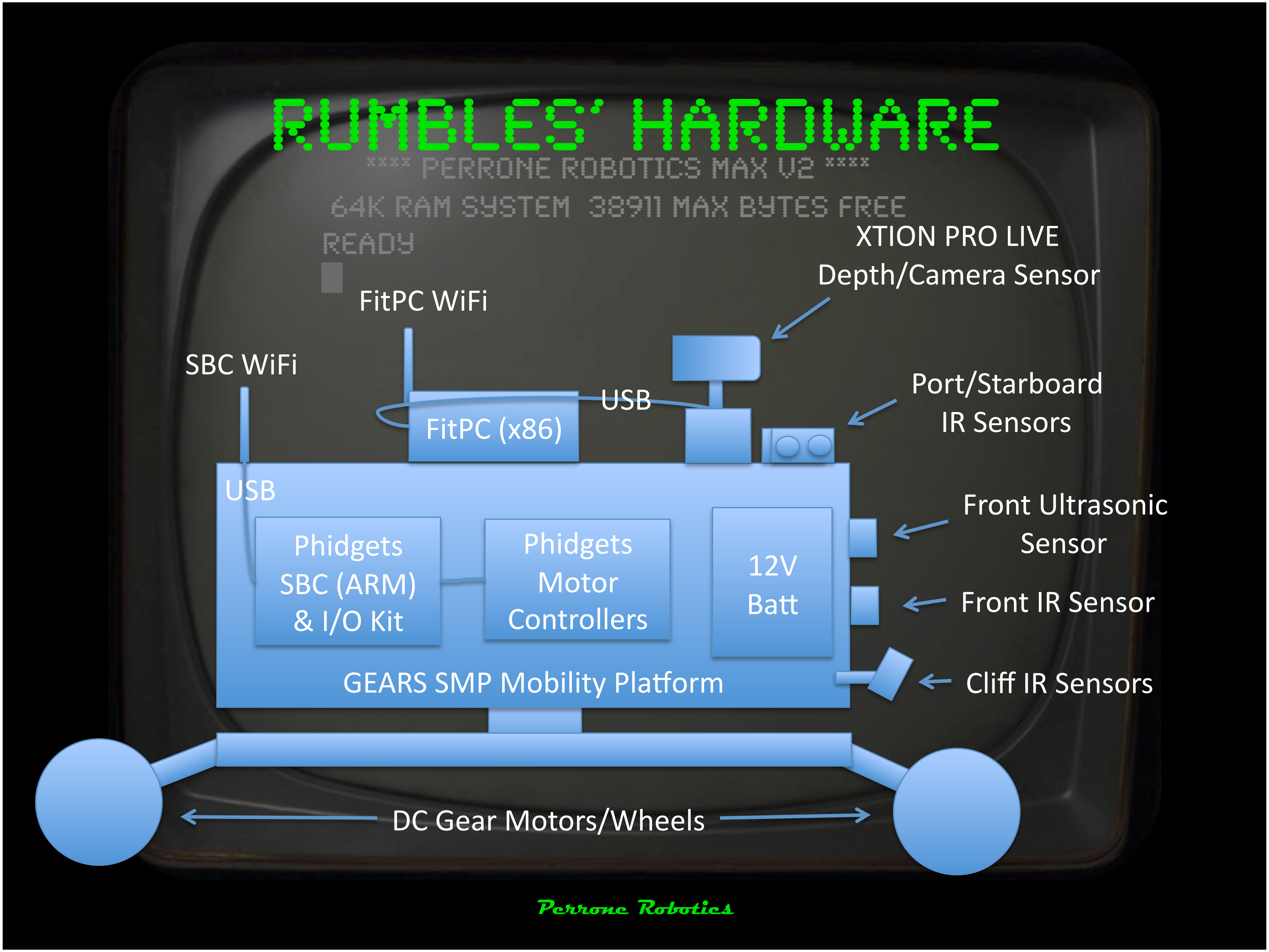
**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- FAMILIES OF SMALL BOTS
  - HOBBYIST, RESEARCH, COMMERCIAL




RUMBLES' HARDWARE
**** PERRONE ROBOTICS MAX V2 ****
64K RAM SYSTEM 38911 MAX BYTES FREE
READY
XTION PRO LIVE
Depth/Camera Sensor
FitPC WiFi
SBC WiFi
USB
FitPC (x86)
Port/Starboard
IR Sensors
USB
Phidgets
SBC (ARM)
& I/O Kit
Phidgets
Motor
Controllers
12V
Batt
Front Ultrasonic
Sensor
Front IR Sensor
GEARS SMP Mobility Platform
Cliff IR Sensors
DC Gear Motors/Wheels
Perrone Robotics

# RUMBLES J1 HOTEL PHOTO SHOOT

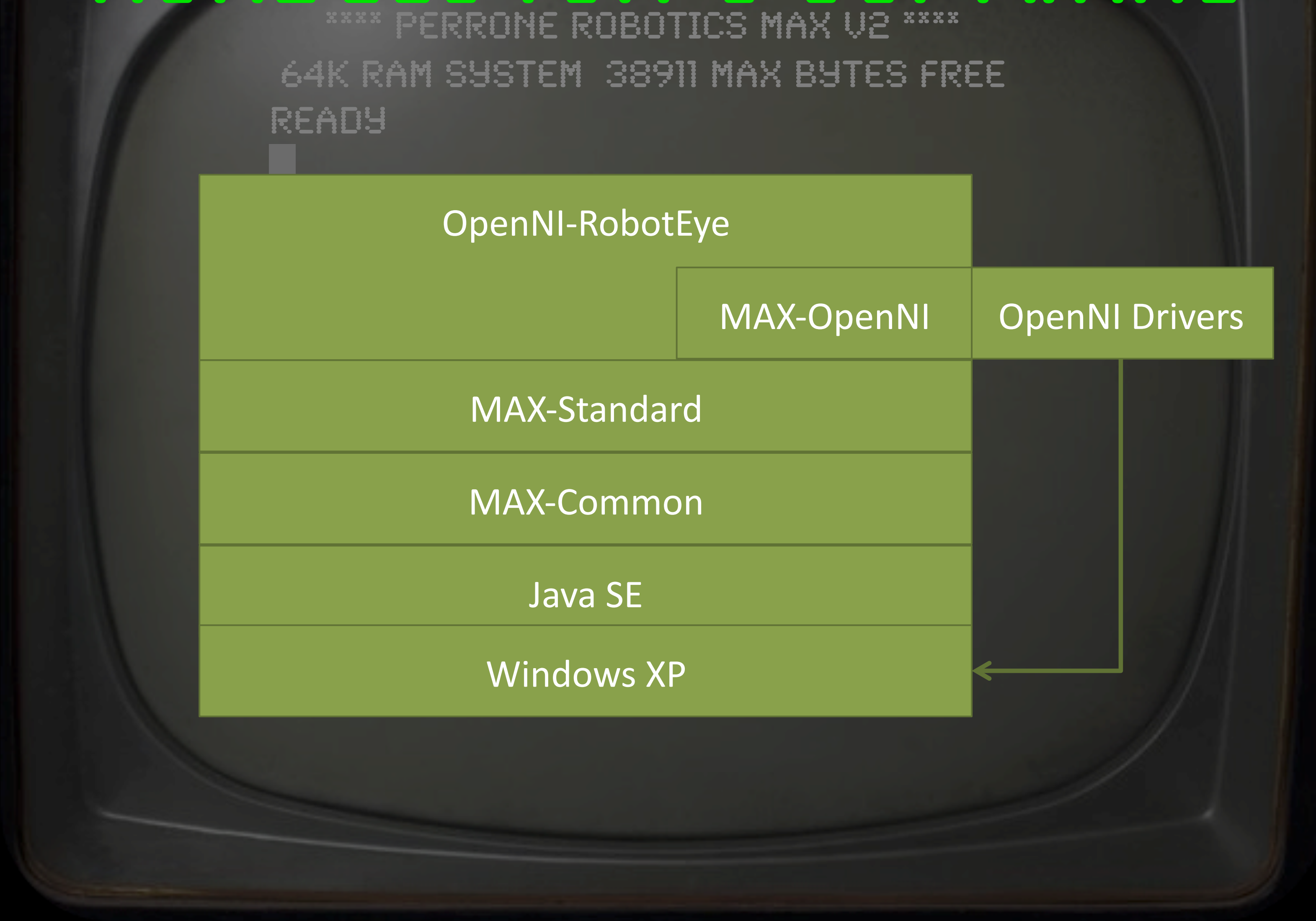
**** PERRONE ROBOTICS MAX V2 ****

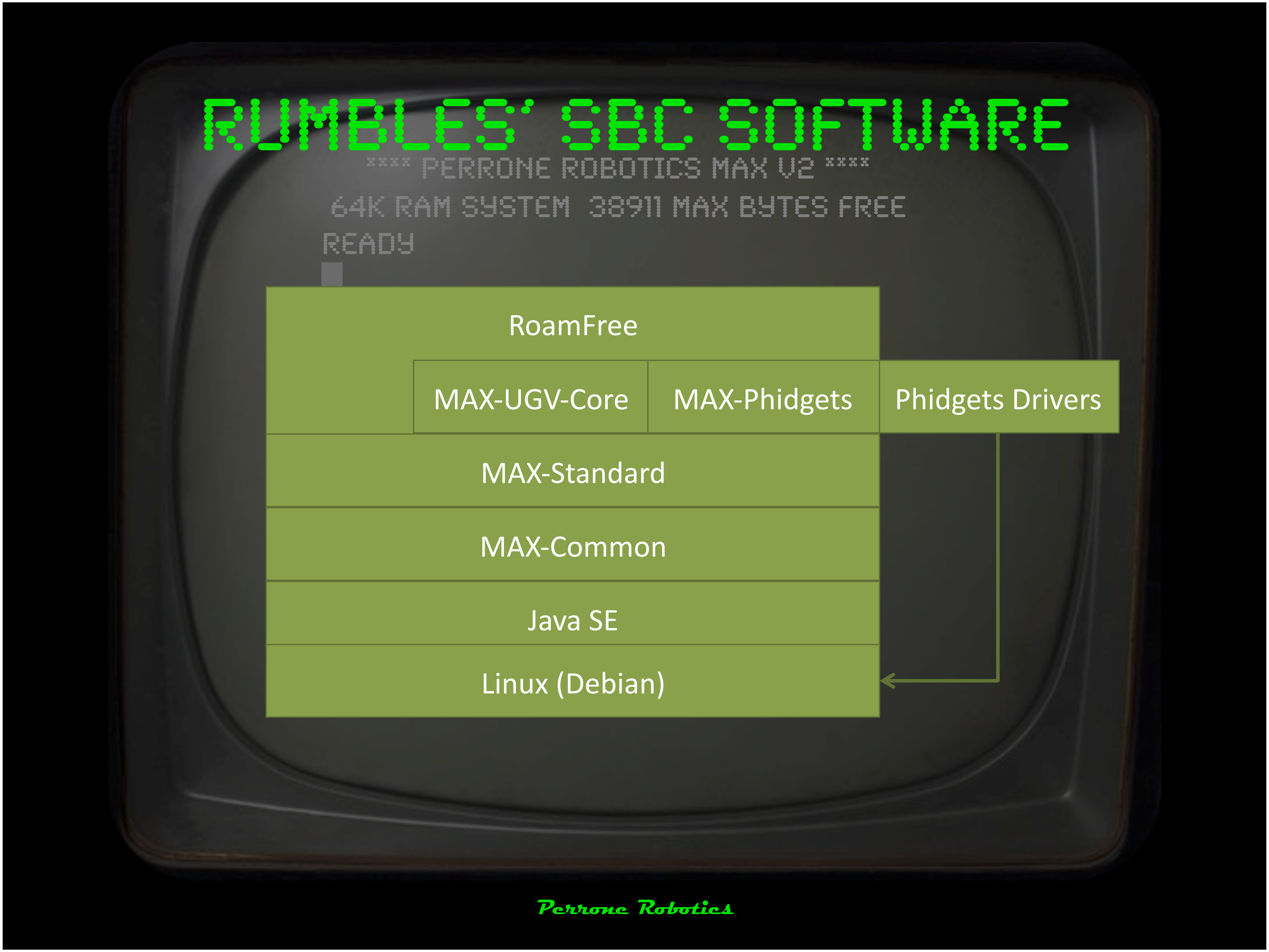
64K RAM SYSTEM  38911 MAX BYTES FREE

READY

Perrone Robotics

# RUMBLES FITPC SOFTWARE




RUMBLES' SBC SOFTWARE
**** PERRONE ROBOTICS MAX V2 ****
64K RAM SYSTEM 38911 MAX BYTES FREE
READY
RoamFree
MAX-UGV-Core
MAX-Phidgets
Phidgets Drivers
MAX-Standard
MAX-Common
Java SE
Linux (Debian)
Perrone Robotics

# EXECUTING A MAX APP

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM  38911 MAX BYTES FREE

READY

**java -classpath <classpath> –Dconfig=./MyConfig MAX**

~ Look for ap/Application.xml in root config

**java -classpath <classpath> MAX**

~ Look for max1.properties with config reference

**java -jar ExecutableJar.jar**

~ Specify classpath and MAX in Manifest.mf

~ Look for max1.properties with config reference

# MAX CONFIG FILE STRUCTURE

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

<u>context/ObjectName.xml</u>

```xml
<?xml version="1.0"?>
<!-- The configuration for a MAX object. Leaf object ID below. -->
<hashtable id=”ObjectName”>
        <!– Object’s class name -->
        <string id="ClassName">com.mypackage.MyClass</string>

        <!– Object reference policy (shared, dedicated, pooled, etc)
        <id id="ReferencePolicy">Shared</id>

        <!– Object specific configuration variables below.
        <hashtable id="Configuration">
                <boolean id=“ActivateThis”>true</boolean>
                <id id=“IDForOtherObjectReferenced>other.context.OtherObject</id>
        </hashtable>
</hashtable>
```

# CONFIGURING AN APP

ap/Application.xml

```
<string id="ClassName">com.perronerobotics.robot.RobotApplication</string>
<id id="SystemID">ap.Robot</id>
```

ap/Robot.xml

```
<string id="ClassName">com.perronerobotics.robot.RobotGeneric</string>
<vector id="ComponentIDs">
        <id id="ComponentIDs">sensors.commandinput.Sensor</id>
        <id id="ComponentIDs">conduct.MotorExcercise</id>
</vector>
```

Generic
MAX
Classes

sensors/commandinput/Sensor.xml

```
<string id="ClassName">com.perronerobotics.sensor.SensorGeneric</string>
<id id="Interface">./StandardInputInterface</id>
<vector id="TriggerAlwaysObjectIDs">
        <id id="TriggerAlwaysObjectIDs">conduct.MotorExcercise</id>
</vector>
```

Next slide

# CONFIGURING AN APP

**** PERRONE ROBOTICS MAX 1.2 ****

## conduct/MotorExcercise.xml

```
<string id="ClassName">com.perronerobotics.popbots.actuation.DCMotorExcercise</string>
<any id="Motors">
        <id id="FR">actuators.motor1.Motor</id>
        <id id="RR">actuators.motor2.Motor</id>
        <id id="FL">actuators.motor3.Motor</id>
        <id id="RL">actuators.motor4.Motor</id>
</any>
```

Custom Extension Class

## actuators/motor1/Motor.xml

```
<string id="ClassName">com.perronerobotics.motor.dc.DCMotorGeneric</string>
<id id="PWM">./PWM</id>
```

Generic MAX Class

## actuators/motor1/PWM.xml

```
<string id="ClassName">com.perronerobotics.motor.phidget.PhidgetMotorControl</string>
<id id="MotorControllerID">target.PhidgetMotorController</id>
<integer id="OutputNumber">0</integer>
```

MAX Phidget Driver Class

Next slide

# CONFIGURING AN APP

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

target/PhidgetMotorController.xml

```
<string id="ClassName">com.perronerobotics.motor.phidget.PhidgetMotorController</string>
<id id="TargetID">target.PhidgetTarget</id>
<integer id="SerialNumber">147641</integer>
```

target/PhidgetTarget.xml

```
<string id="ClassName">com.perronerobotics.phidget.PhidgetTarget</string>
<string id="IPAddress">10.0.0.61</string>
<integer id="Port">5001</integer>
```

MAX
Phidget
Driver
Classes

# CONTROLLING MOTORS

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

```
public class DCMotorExcercise extends com.perronerobotics.plan.PlanGeneric{

 public static final ID MOTORS = ID.id("Motors");
 private ObjectMap motors = null;

 public void configure(Configuration config) {
      // Configure a map of motor objects
      motors = new ObjectMap();
      motors.configure(config, MOTORS);
      …
 }
```

*For simplicity, exception handling, comments, and non critical code left out*

# CONTROLLING MOTORS

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM  38911 MAX BYTES FREE

READY

```
public void trigger(ID eventType, Info info) {
    if (info instanceof SignalsString) {
        // Parse string to get a command (e.g. RUN, motor ID, and rate)
        if(type.equalsIgnoreCase("RUN")){
            ID motorID = // parse from command
            double rate = // parse from command

            DCMotor motor = (DCMotor) motors.getObject(motorID);
            motor.run(rate); // rate = -1.0 to 0.0 to 1.0
        }
        else ....
    }
}
```

# CONTROLLING MOBILITY

**** PERRONE ROBOTICS MAX V2 ****

READY

```
public class MobilizationPlanGeneric {

  public static final ID MOBILITY_PLATFORM = ID.id("MobilityPlatform");
  private SkidSteerPlatform mobilityPlatform = null;


  public void configure(Configuration config) {
        mobilityPlatform
        = (SkidSteerPlatform) ObjectRegistry.getObject(config, MOBILITY_PLATFORM);
        …
   }

// Example calls
mobilityPlatform.move(rate);
mobilityPlatform.rotate(rate);
mobilityPlatform.move(portRate, starboardRate);
mobilityPlatform.stop();
```

# MOBILITY CONFIGURATION

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- conduct/Mobilization.xml

```
<string id="ClassName">
    com.perronerobotics.popbots.actuation.mobility.Mobilization</string>
<id id="MobilityPlatform">ugv.mobility.MobilityPlatform</id>
```

- ugv/mobility/MobilityPlatform.xml

```
<string id="ClassName">
    com.perronerobotics.mobility.wheeled.skidsteer.SkidSteerPlatformGeneric</string>
<any id="SkidSteerMap">
    <vector id="PortSide">
        <id id="Actuators">./actuators.motor3.Motor</id>
        <id id="Actuators">./actuators.motor4.Motor</id>
    </vector>
    <vector id="StarboardSide">
        <id id="Actuators">./actuators.motor1.Motor</id>
        <id id="Actuators">./actuators.motor2.Motor</id>
    </vector>
</any>
```

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM  38911 MAX BYTES FREE

READY

...THAT'S ACTUATION

(MOTORING AND MOBILITY)

MOVING ONTO SENSATION...

# SENSATION CONFIGURATION

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- ugv/mobility/sensation/ObjectDetection.xml

```
<string id="ClassName">
    com.perronerobotics.physical.sense.ObjectSensationPerspectives</string>
<any id="Views">
    <id id="FrontView">./FrontView</id>
    <id id="CliffView">./CliffView</id>
    <id id="PortView">./PortView</id>
    <id id="StarboardView">./StarboardView</id>
</any>
```

- ugv/mobility/sensation/FrontView.xml

```
<string id="ClassName">
    com.perronerobotics.popbots.obstacles.RobotRelativeObstacleDetection</string>
<any id="AnalogInputs">
    <id id="IRF">./sensors.sensor1.Sensor</id>
    <id id="US">./sensors.sensor6.Sensor</id>
</any>
```

# SENSOR CONFIGURATION

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- ugv/mobility/sensation/sensors/sensor1/Sensor.xml

```
<string id="ClassName">com.perronerobotics.analog.AnalogInputGeneric</string>
<id id="Interface">./PhidgetAnalogInput</id>
```

- ugv/mobility/sensation/sensors/sensor1/PhidgetAnalogInput.xml

```
<string id="ClassName">
     com.perronerobotics.analog.phidget.PhidgetSBCAnalogInput</string>
<id id="InterfaceKitID">target.PhidgetInterfaceKit</id>
<integer id="InputNumber">0</integer>
<boolean id="Inverse">true</boolean>
<double id="Multiplier">94620</double>
<double id="Offset">-16.92</double>
<double id="MinInputValue">80</double>
<double id="MaxInputValue">490</double>
```

# SENSATION

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

```
public class RobotRelativeObstacleDetection extends
                    com.perronerobotics.physical.sense.ObjectDetectorPerspective {

 public static final ID ANALOG_INPUTS = ID.id("AnalogInputs");
  protected ObjectMap analogInputs = new ObjectMap();

  public void configure(Configuration factors){
       analogInputs.configure(config, ANALOG_INPUTS);
       …
  }

   protected void startSensation(Info info){
            sensedObjects.clear();
   }
```

# SENSATION

**** PERRONE ROBOTICS MAX V2 ****

```
protected void senseObjects(Info info) {
    Enumeration analogInputKeysIt = analogInputs.keys();
    while (analogInputKeysIt.hasMoreElements()) {
        ID analogInputID = (ID) analogInputKeysIt.nextElement();

        AnalogInput analogInput
            = (AnalogInput) analogInputs.getObject(analogInputID);

        double analogValue = analogInput.readValue();

        getRobotRelativeObjects().associateReading((long) analogValue);
    }
}
```

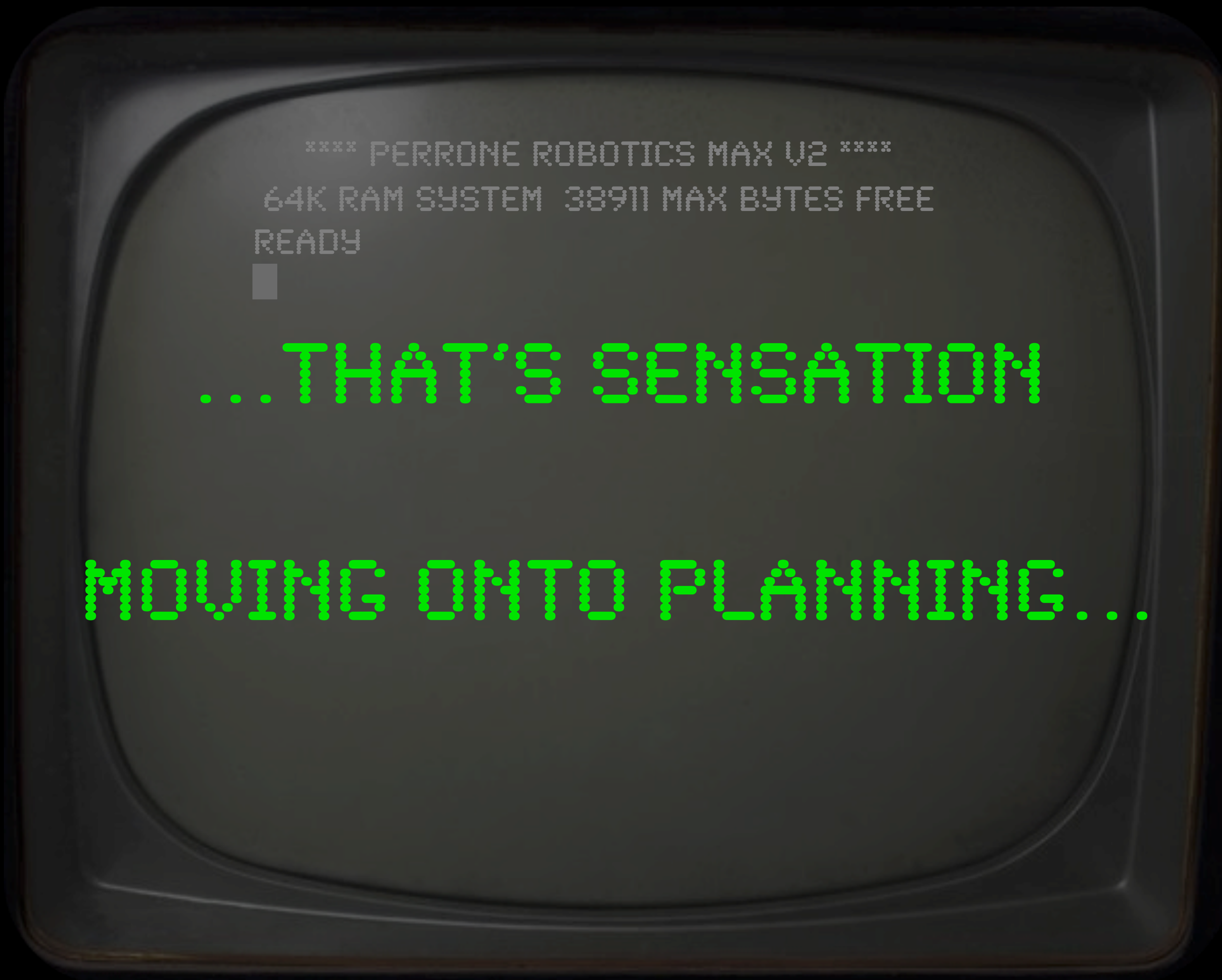
# SIMPLE OBSTACLE DETECTION

```
public class RobotRelativeObstacles
        extends com.perronerobotics.physical.object.PhysicalObjects {

    public void clear(){
        collisionDistance = Long.MAX_VALUE;
        collisionAngle = 0;
        maxObstacleDistance = 0;
    }

    public void associateReading(long distance){
        if(distance < collisionDistance){
            collisionDistance = distance;
        }

        if(distance > maxObstacleDistance){
            maxObstacleDistance = distance;
        }
    }
    …
}
```


**** PERRONE ROBOTICS MAX V2 ****
64K RAM SYSTEM 38911 MAX BYTES FREE
READY
...THAT'S SENSATION
MOVING ONTO PLANNING...

# ROAM FREE APPLICATION

- SIMPLE EXAMPLE ILLUSTRATING ROBOT MOVEMENTS
- SENSE ENVIRONMENT
- DETECT COLLISION DISTANCES/ANGLES
- EXECUTE SPECIFIC MANEUVERS BASED ON SENSED DISTANCES/ANGLES
- MANUEVERS ARE SEQUENCES OF ONE OR MORE MANEUVER ACTIONS
- PRIORITIZE WHICH MANEUVERS ARE EXECUTED

# ROAMFREE CONFIGURATION

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- ap/Application.xml

```
<string id="ClassName">com.perronerobotics.robot.RobotApplication</string>
<id id="SystemID">ap.Robot</id>
<any id="SystemThreadIDs">
    <vector id="ap.SynchThread">
        <id id="ap.SynchThread">ugv.MovementPlanner</id>
    </vector>
</any>
```

- ap/SynchThread.xml

```
<string id="ClassName">com.perronetech.ap.SynchronousSystemThread</string>
<integer id="SynchPauseTime">25</integer>
```

# ROAMFREE CONFIGURATION

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- ap/Robot.xml

<string id="ClassName">com.perronerobotics.robot.RobotGeneric</string>

<vector id="ComponentIDs">

    <id id="ComponentIDs">ugv.sensation.sensors.commandinput.Sensor</id>

    <id id="ComponentIDs">ugv.EstopHandler</id>

    **<id id="ComponentIDs">ugv.MovementPlanner</id>**

    <id id="ComponentIDs">ugv.sensation.sensors.opennisensor.Peripheral</id>

</vector>

# ROAM FREE MOVEMENT PLAN

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- HERE IS WHAT WE WANT ALGORITHMICALLY...
  - IF BUMP OR IMPACTFUL IN FRONT
    - BACK UP A BIT, SPIN AROUND, MOVE ALONG
  - ELSE IF COLLIDABLE IN FRONT/SIDES
    - TRIGGER A ROTATION PROPORTIONAL TO COLLISION ANGLE
  - ELSE
    - ROAM AROUND TRIGGERING SLIGHT ADJUSTMENTS ALONG WAY AS SEE THINGS IN DISTANCE
    - ADJUST SPEED PROPORTIONAL TO COLLISION DISTANCES
    - ADJUST TURNING PROPORTIONAL TO COLLISION ANGLES

# MAX-UGV Movement Planning

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- If in run/enable mode:
  - Induce any sensation updates
  - Ask maneuvers to detect if active
  - Resolve and prioritize manuevers to execute
  - Ask the remaining active manuevers to execute
- Else:
  - Ensure mobility platform stopped

# MOVEMENT PLANNER CONFIG

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- ugv/MovementPlanner.xml

```
<string id="ClassName">com.perronerobotics.movement.MovementPlannerGeneric</
    string>
<id id="MobilityPlatformID">./mobility.MobilityPlatform</id>
<id id="ObjectDetectors">./sensation.ObjectDetection</id>
<vector id="Maneuvers">
    <id id="Maneuvers">./maneuvers.ImpactAvoidance</id>
    <id id="Maneuvers">./maneuvers.CliffAvoidance</id>
    <id id="Maneuvers">./maneuvers.CollisionAvoidance</id>
    <id id="Maneuvers">./maneuvers.Roaming</id>
    <id id="Maneuvers">./maneuvers.IRRoaming</id>
</vector>
<boolean id="OrderedPriorityConsideration">true</boolean>
```

# MANEUVER CONFIGURATION

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

- ugv/maneuvers/ImpactAvoidance.xml

```
<string id="ClassName">
     com.perronerobotics.popbots.maneuver.ImpactAvoidanceManeuver</string>

<!-- Min collision distance for this maneuver to trigger active -->
<long id="MinCollisionDistance">250</long>

<!--  Actions for this maneuver -->
<vector id="Actions">
     <id id="Actions">./StopRobot</id>
     <id id="Actions">./RollBack</id>
     <id id="Actions">./Spin</id>
</vector>
```

# BASE AVOIDANCE MANEUVER

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

```
public class AvoidanceManeuver
	extends com.perronerobotics.movement.ManeuverGeneric {

	public static final ID FRONT_OBSTACLES = ID.id("FrontView");
	public static final ID CLIFF_OBSTACLES = ID.id("CliffView");
	public static final ID PORT_OBSTACLES = ID.id("PortView");
	public static final ID STARBOARD_OBSTACLES = ID.id("StarboardView");

	public void configure(Configuration factors){
		super.configure(factors);

		minCollisionDistance
			= factors.getLong(MIN_COLLIDE_DISTANCE, minCollisionDistance);
		maxCollisionDistance
			= factors.getLong(MAX_COLLIDE_DISTANCE, maxCollisionDistance);
	}
	…
}
```

# IMPACT AVOIDANCE MANEUVER

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

```
public class ImpactAvoidanceManeuver extends AvoidanceManeuver
        implements SelfExitingManeuver{

        public boolean detectIfActive(){
                RobotRelativeObstacles frontViewObstacles
                        = (RobotRelativeObstacles) super.getPhysicalObjects(FRONT_OBSTACLES);

                long frontDistance = frontViewObstacles.getCollisionDistance();

                if(frontDistance < minCollisionDistance){
                        collisionAngle = 0;
                        collisionDistance = frontDistance;
                        return true;
                }else{
                        return false;
                }
        }
}
```

# STOP ROBOT ACTION

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- ugv/maneuvers/StopRobot.xml

  <string id="ClassName”>com.perronerobotics.popbots.maneuver.actions.StopRobot</string>

```
public class StopRobot extends com.perronerobotics.movement.ManeuverAction {

  public boolean execute() {
        super.getMobilityPlatform().stop();

         return true;
   }
}
```

# ROLL BACK ACTION

- ugv/maneuvers/RollBack.xml

```
<string id="ClassName”>com.perronerobotics.popbots.maneuver.actions.RollBack</string>
<double id="ActionTimeoutSeconds">2.0</double>
<double id="BackupSpeed">0.3</double>
```

```
public class RollBack extends com.perronerobotics.movement.TimedManeuverAction {
    public static final ID BACKUP_SPEED = ID.id("BackupSpeed");
    private double rollBackSpeed = 0.3;

    public void configure(Configuration factors){
        super.configure(factors);
        rollBackSpeed = factors.getDouble(BACKUP_SPEED, rollBackSpeed);
    }
    protected void executeIteration(){
      ((BiDirectionalWheeledPlatform) super.getMobilityPlatform()).move(-rollBackSpeed);
    }
    protected void executeTimeout() {
        super.getMobilityPlatform().stop();
    }}
```

# SPIN ACTION

- ugv/maneuvers/Spin.xml

```
<string id="ClassName”>com.perronerobotics.popbots.maneuver.actions.Spin</string>
<double id="ActionTimeoutSeconds”>3.0</double>
<double id=”SpinSpeed">0.7</double>
```

```
public class Spin extends com.perronerobotics.movement.TimedManeuverAction {
    public static final ID SPIN_SPEED = ID.id("SpinSpeed");
    protected double spinSpeed = 0.3;

    public void configure(Configuration factors){
        super.configure(factors);
        spinSpeed = factors.getDouble(SPIN_SPEED, spinSpeed);
    }
    protected void executeIteration(){
        ((SkidSteerPlatform) super.getMobilityPlatform()).rotate(spinSpeed);
    }
    protected void executeTimeout() {
        super.getMobilityPlatform().stop();
    }}
```

# OTHER MANEUVERS

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- CliffAvoidance
  - MaxCollisionDistance = 550 mm // Triggers if cliff sensors exceed this
  - Actions = StopRobot, RollBack, Spin
- CollisionAvoidance
  - MinCollisionDistance = 450 mm // Triggers if side IR sensors less than this
  - Actions = SpinProportionally
- Roaming
  - Ethernet connection to MAX OpenNI-RobotEye server on FitPC
  - Gets collection of Features
  - Looks for Openings and modulates speed
  - Actions = Roam
- IRRoaming
  - Uses side IR sensors for roaming (if Feature Roaming fails)
  - Actions = PointRoam

# Feature Roam Configuration

- ugv/maneuvers/Roaming.xml

```
<string id="ClassName">
    com.perronerobotics.popbots.maneuver.FeatureRoamingManeuver</string>
<id id="FeatureSensor">../sensation.sensors.opennisensor.Peripheral</id>
<vector id="Actions">
    <id id="Actions">./Roam</id>
</vector>
```

- ugv/sensation/sensors/opennisensor/Peripheral.xml

```
<string id="ClassName">com.perronerobotics.sensor.SensorGeneric</string>
<id id="Interface">./EthernetClientPort</id>
<vector id="TriggerAlwaysObjectIDs">
    <id id="TriggerAlwaysObjectIDs">../../../maneuvers.Roaming</id>
</vector>
<!– Configuration info on how to packetize incoming data →
```

- ugv/sensation/sensors/opennisensor/EthernetClientPort.xml

```
<string id="ClassName">
    com.perronetech.comm.ethernet.EthernetClientPortGeneric</string>
<string id="IPAddress">10.0.0.62</string>
<integer id="PortNumber">10101</integer>
```

# FEATURE ROAM MANEUVER

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

```
public class FeatureRoamingManeuver extends AvoidanceManeuver{

    public synchronized void trigger(ID eventID, Info info){
        if(info instanceof Features){
            features = (Features) info;
    }

    …
```

# FEATURE ROAM MANEUVER

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

```
public synchronized boolean detectIfActive(){
        for(int i=0; i < features.size(); ++i){
                Feature feature = (Feature) features.getObject(i);
                if(feature.isOpening()){
                        double centerAngle = feature.getCenterAngle();
                        if(Mathematics.abs(centerAngle) < targetAngle){
                                opening = feature;
                                targetAngle = centerAngle;
                        }
                }

                int distance = feature.getClosestDistance();
                if(distance < collisionDistance){
                        collisionDistance = distance;
                }
        }
        …
```

# FEATURE ROAM MANEUVER

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

```
…
this.setCollisionDistance(collisionDistance);

if(opening == null){
    int minWidth = -1;
    for(int i=0; i < features.size(); ++i){
        Feature feature = (Feature) features.getObject(i);
        if(!feature.isBump()){
            double centerAngle = feature.getCenterAngle();
            int width = feature.getWidth();
            if(width > minWidth){
                dent = feature;
                targetAngle = centerAngle;
                minWidth = width;
            }
        }
    }
}
…
```

# FEATURE ROAM MANEUVER

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

```
    …
    if(dent == null){
        targetAngle = 0;
    }

    this.setCollisionAngle(targetAngle);

    return true;
}else{
    return false;
}
}
}
```

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

# RUMBLES DURING DEVELOPMENT: SOME VIDEOS FOR YOUR VIEWING PLEASURE WHILE RUMBLES' ROAM FREE DEMO GETS SETUP

# VIDEO #1: MAKING THE BABY CRY

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM  38911 MAX BYTES FREE

# VIDEO #2: RUMBLES ROAMING

**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM  38911 MAX BYTES FREE

# VIDEO #3: LUCA HATES ON DADDY'S ROBOT BETA

64K RAM SYSTEM  38911 MAX BYTES FREE

READY

# More Information

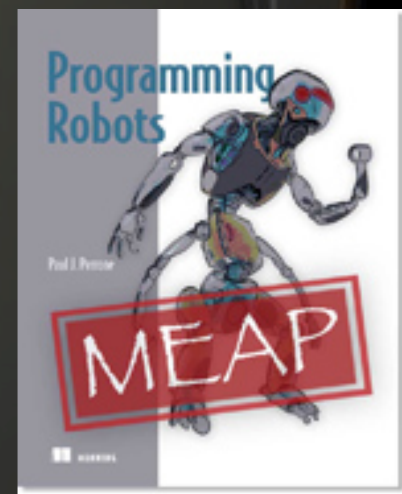
**** PERRONE ROBOTICS MAX V2 ****

64K RAM SYSTEM 38911 MAX BYTES FREE

READY

- Perrone Robotics:
  - www.perronerobotics.com
  - paul@perronerobotics.com
  - 703-728-0115
- Robot Kits/Solutions
  - Contact me to inquire about kits (unassembled or pre-assembled) and solutions

Robot Book
Coming Soon




http://www.manning.com/perrone/