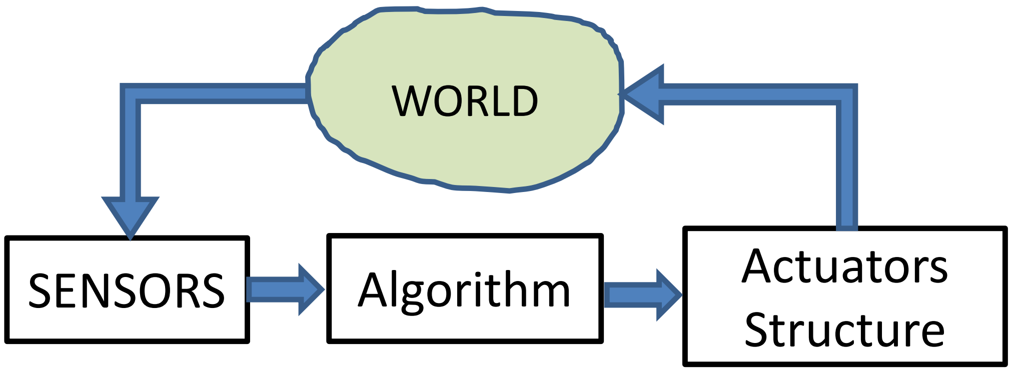
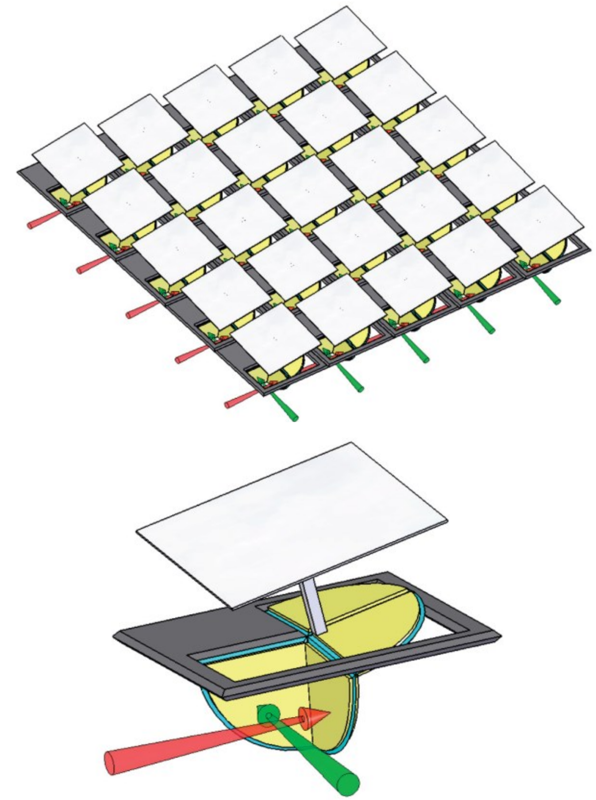
EECS192 Lecture 1
Jan. 21, 2020

- What is Mechatronics?
- Project Description
- Autonomous system example
- Course Organization
- ARM Cortex A8 overview
- Pthreads and timing
- Electronic Components

What is Mechatronics?

- Moore’s Law for electronics
- Moore’s Law for mechanics(?)





Folded mirror array

Key Technologies for Mechatronics:

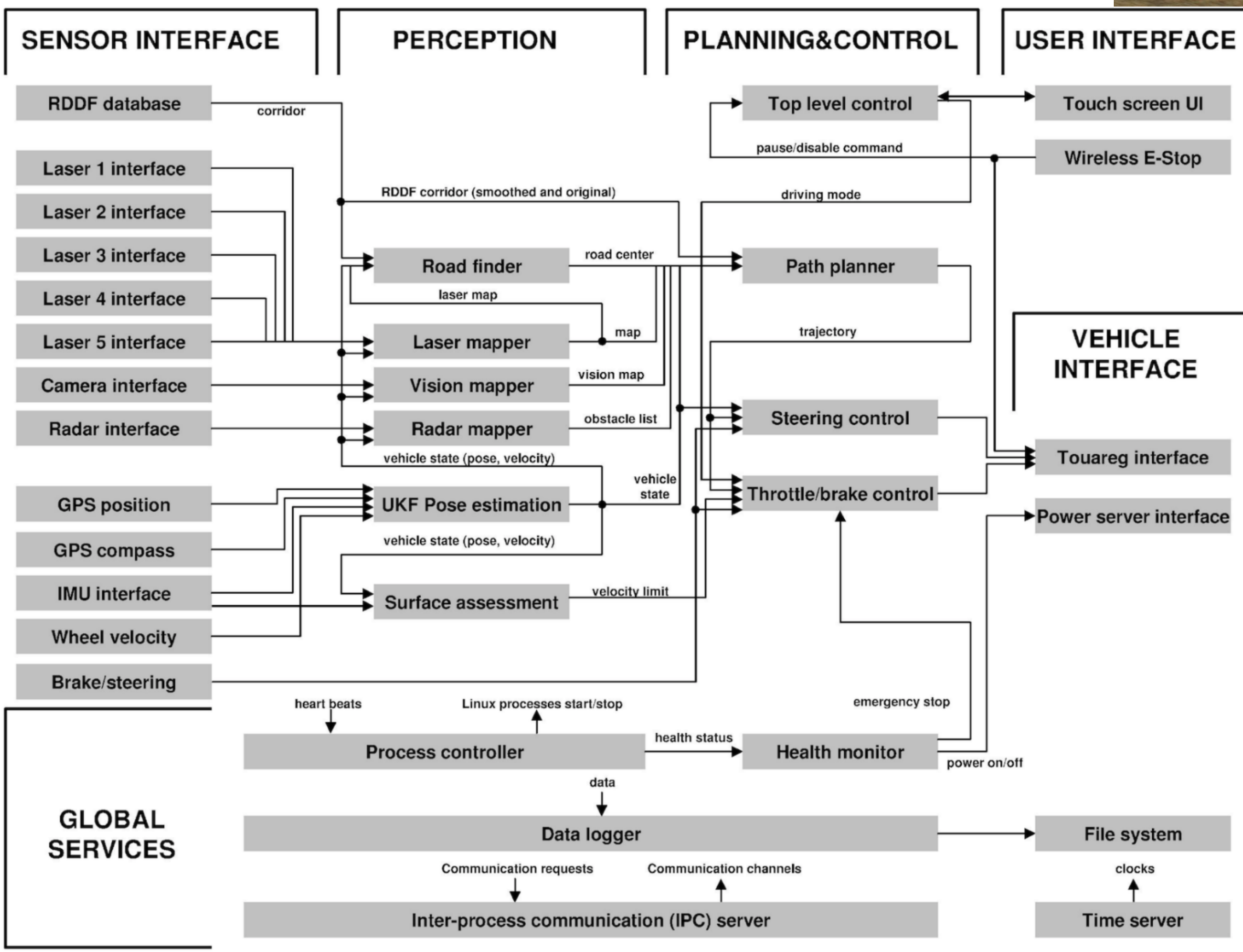
- Signal processing
- Control
- …

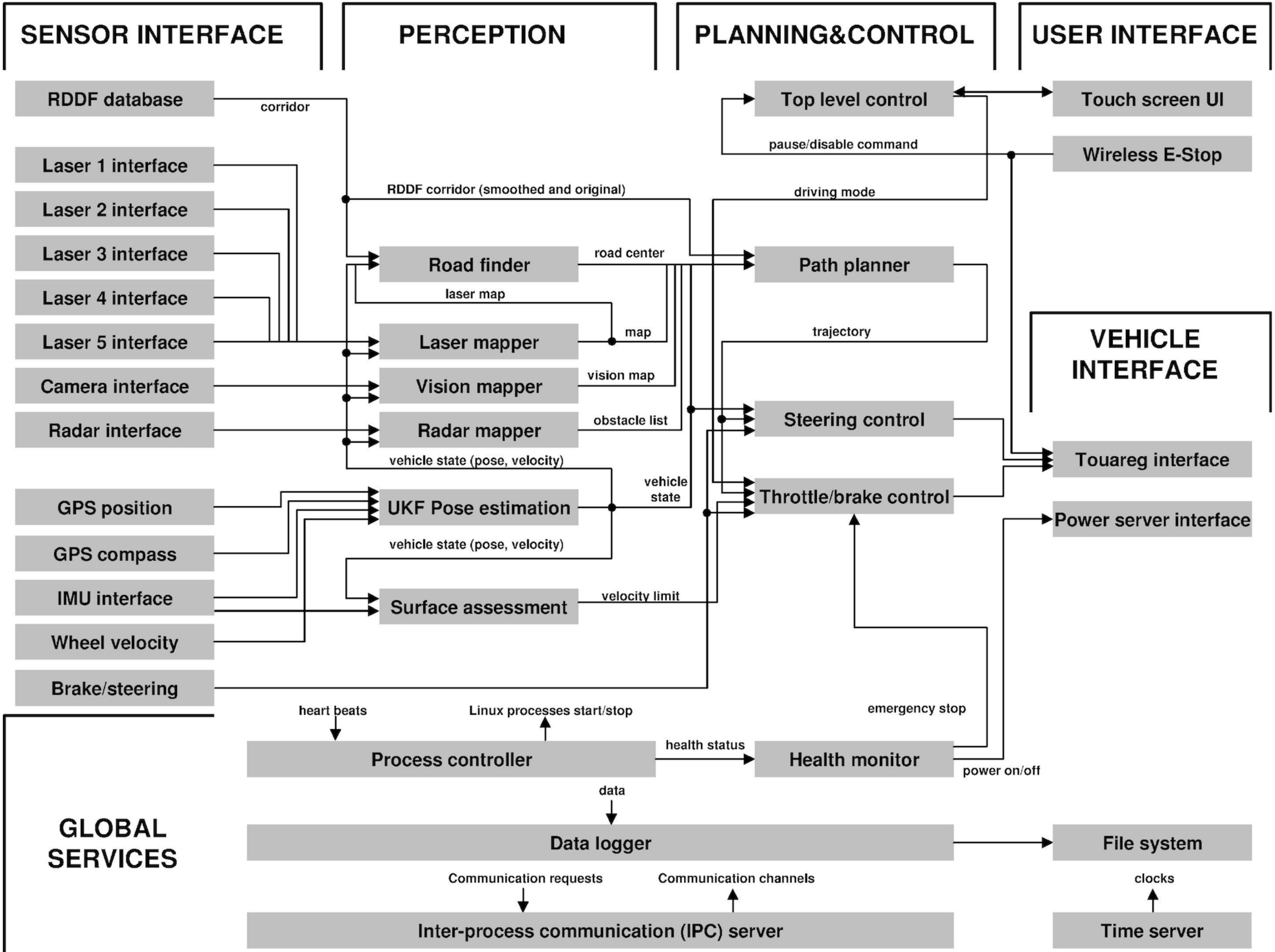
Project Description

- Design Autonomous Race Car
- Unknown track
- Follow track without hitting cones.
- Stop at end of track.
- Winning speed: 3.3 m/sec (Spring 2017 Natcar winner)
- (2018 9.7 ft/sec = 3.0 m/sec)
- Learning allowed (though only have 5 minutes total for best run)

Hardware

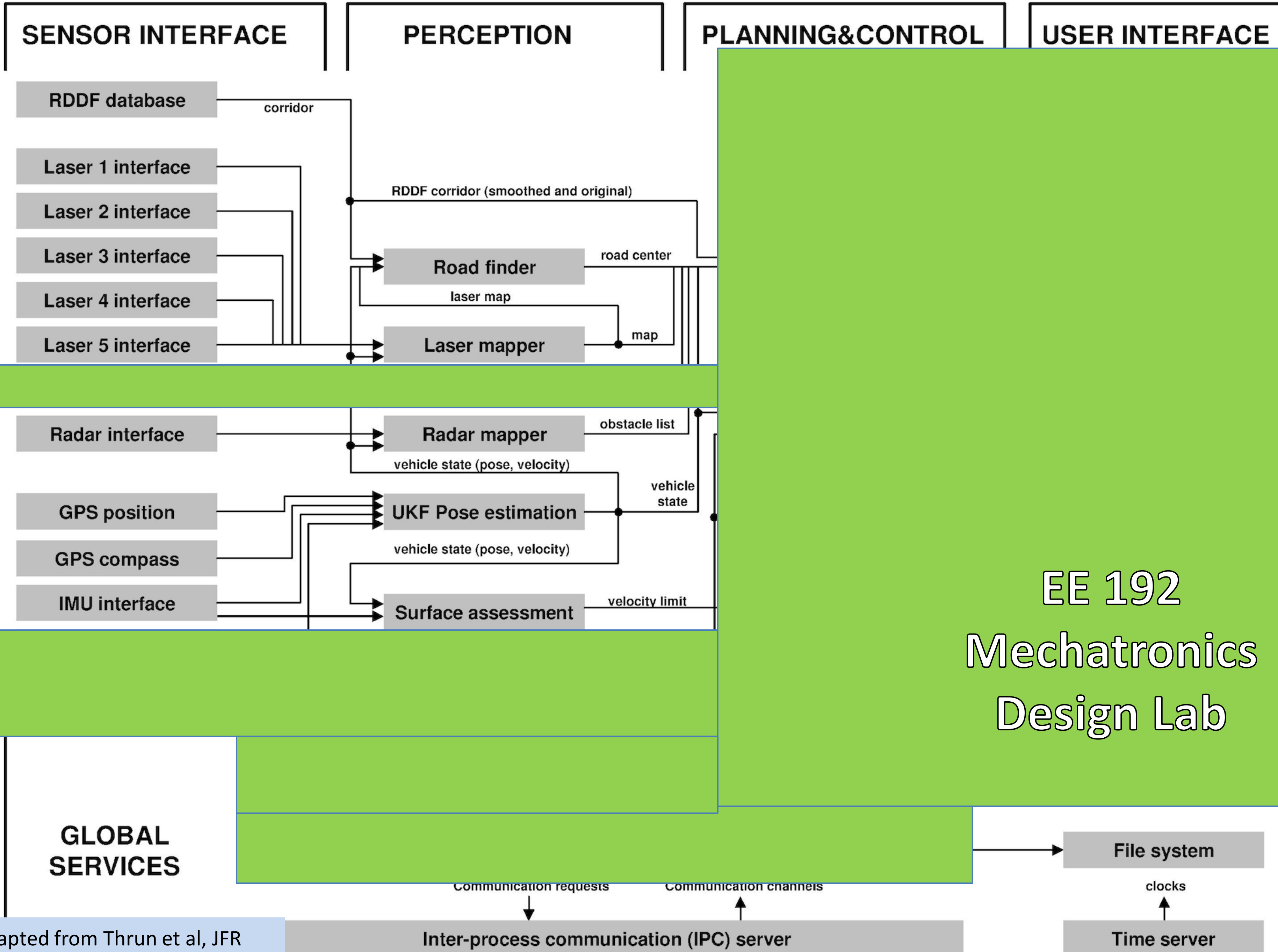
Thrun et al Stanley 2005



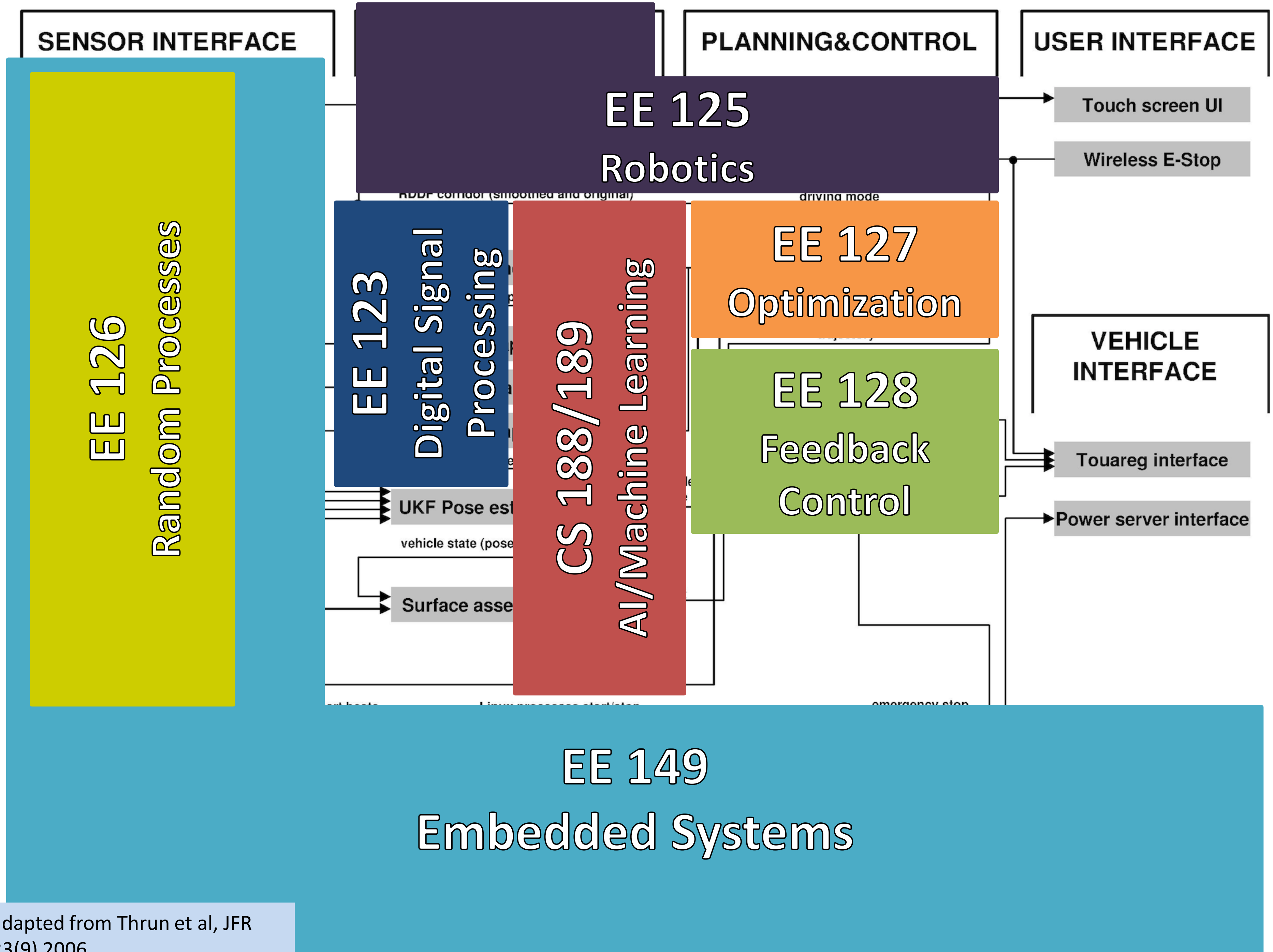


SENSOR INTERFACE
PERCEPTION
PLANNING&CONTROL
USER INTERFACE
RDDF database
corridor
Top level control
Touch screen UI
pause/disable command
Wireless E-Stop
Laser 1 interface
Laser 2 interface
Laser 3 interface
Laser 4 interface
Laser 5 interface
Camera interface
Radar interface
RDDF corridor (smoothed and original)
driving mode
Road finder
road center
Path planner
laser map
map
trajectory
VEHICLE INTERFACE
Laser mapper
Vision mapper
vision map
Radar mapper
obstacle list
Steering control
vehicle state (pose, velocity)
Touareg interface
vehicle state
GPS position
UKF Pose estimation
Throttle/brake control
GPS compass
Power server interface
vehicle state (pose, velocity)
IMU interface
Surface assessment
velocity limit
Wheel velocity
Brake/steering
heart beats
Linux processes start/stop
emergency stop
Process controller
health status
Health monitor
power on/off
data
GLOBAL SERVICES
Data logger
File system
Communication requests
Communication channels
clocks
Inter-process communication (IPC) server
Time server

EE 192 Mechatronics Design Lab



adapted from Thrun et al, JFR 23(9) 2006



adapted from Thrun et al, JFR 23(9) 2006

Course Organization

- Tues Lecture -> Wed Lab Demo -> 9 days -> Fri checkoff hour (tba)
- Partners: 3 ideal. Only have 12 teams, so priority to teams of 3. $300 deposit for car+CPU+battery+camera+motor+servo
- Checkoffs ``better is enemy of good''- robustness: needs to work in a window

Course Organization (cont)

Checkpoint sequence:

- *OLD: CPU -> drive motor+servo -> power supply -> line sense -> line follow/fig 8 -> velocity control ->*
- NEW: CPU -> drive motor+servo -> line sense -> line follow/fig 8 -> velocity control -> power+motor PCB ->

Round 1/CalDay/Round2/
NatCAR (Sat May 9, 2020)

Backup motor drive+power supply hardware for safety:

Race 1: max 8/10   Race 2: Max 15/20

Course Organization (cont)

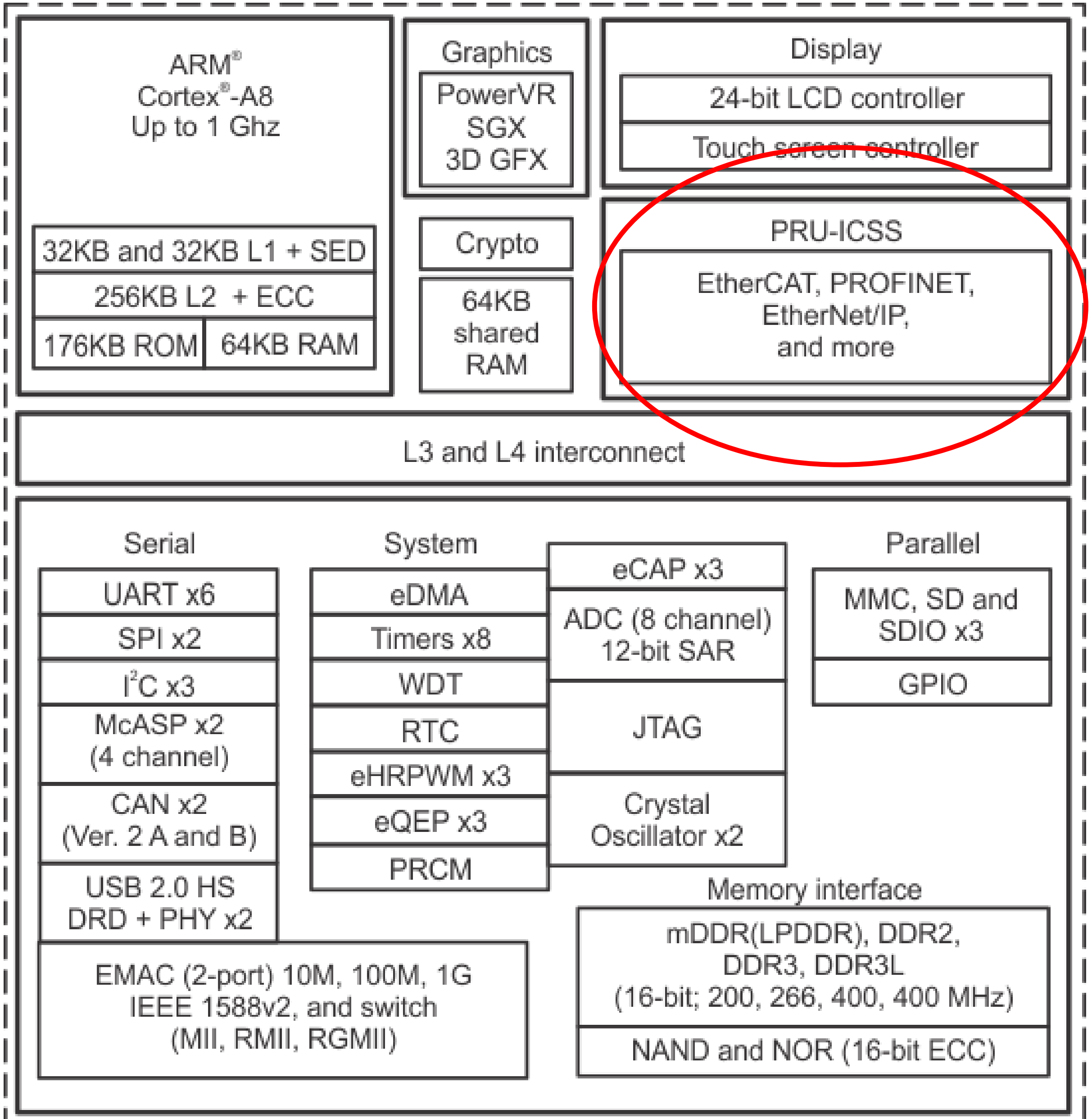
- Emphasis: robustness, simplicity. Design/Simulate/Build/Test
- Goal: 10 hours per week per team member.
  - Part of checkpoint: report weekly hours (important for course tuning)

- What about Complexity?
- Reliability of the overall system $(90\%)^N$

(connectors, power supply, heat sinks, solder joints, CPU stack, car mechanics, camera mount, control stability, lighting robustness,…)

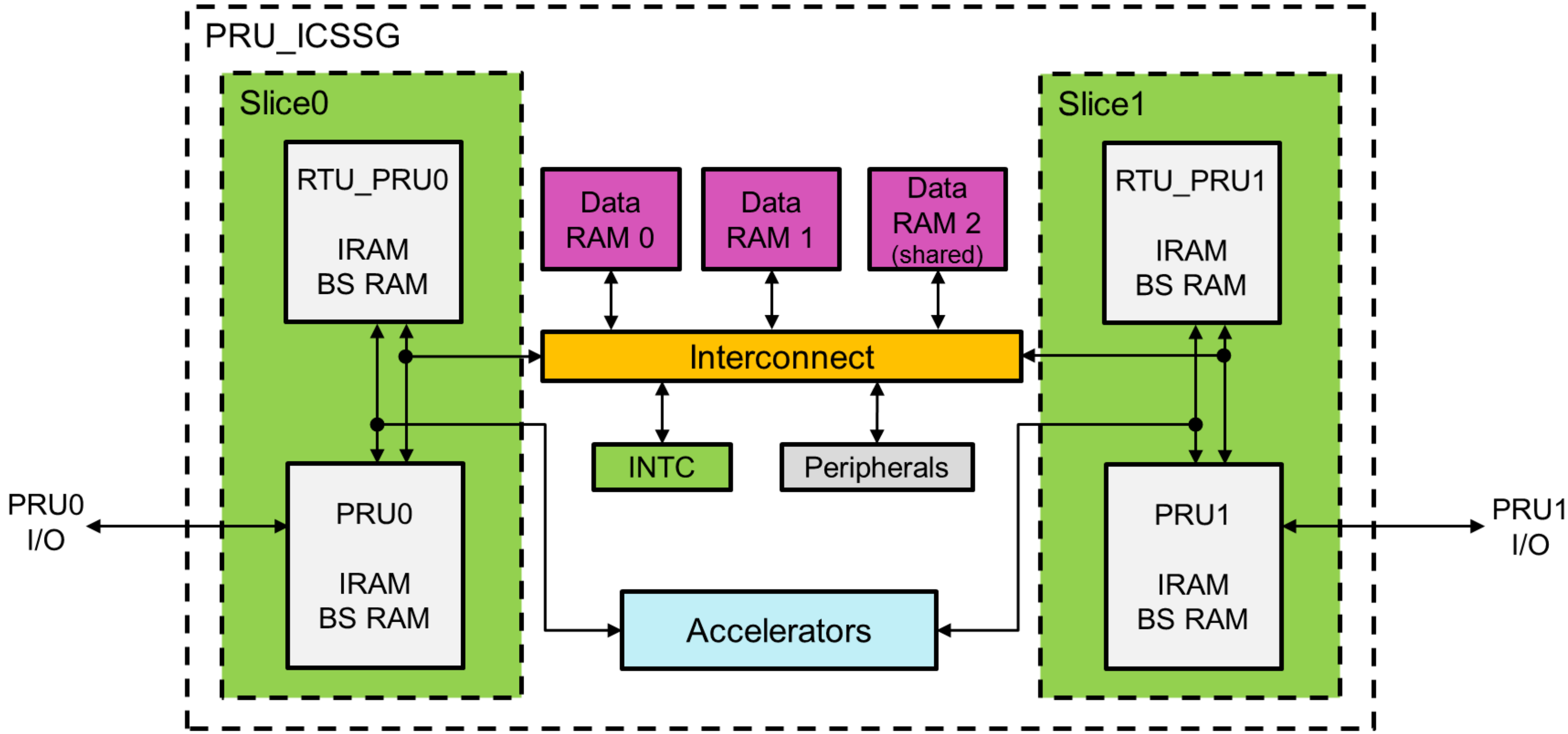
ARM Cortex A8 Overview

ARM Cortex A8 Overview



Beaglebone Blue
TI **Sitara™ AM335x Processors**
ARM Cortex A8
4GB Flash
512 MB RAM
32 bit ARM 7 core
1 GHz
A/D
3x SPI
Timers
WiFi
USB
microSD card

Programmable Real-time Unit

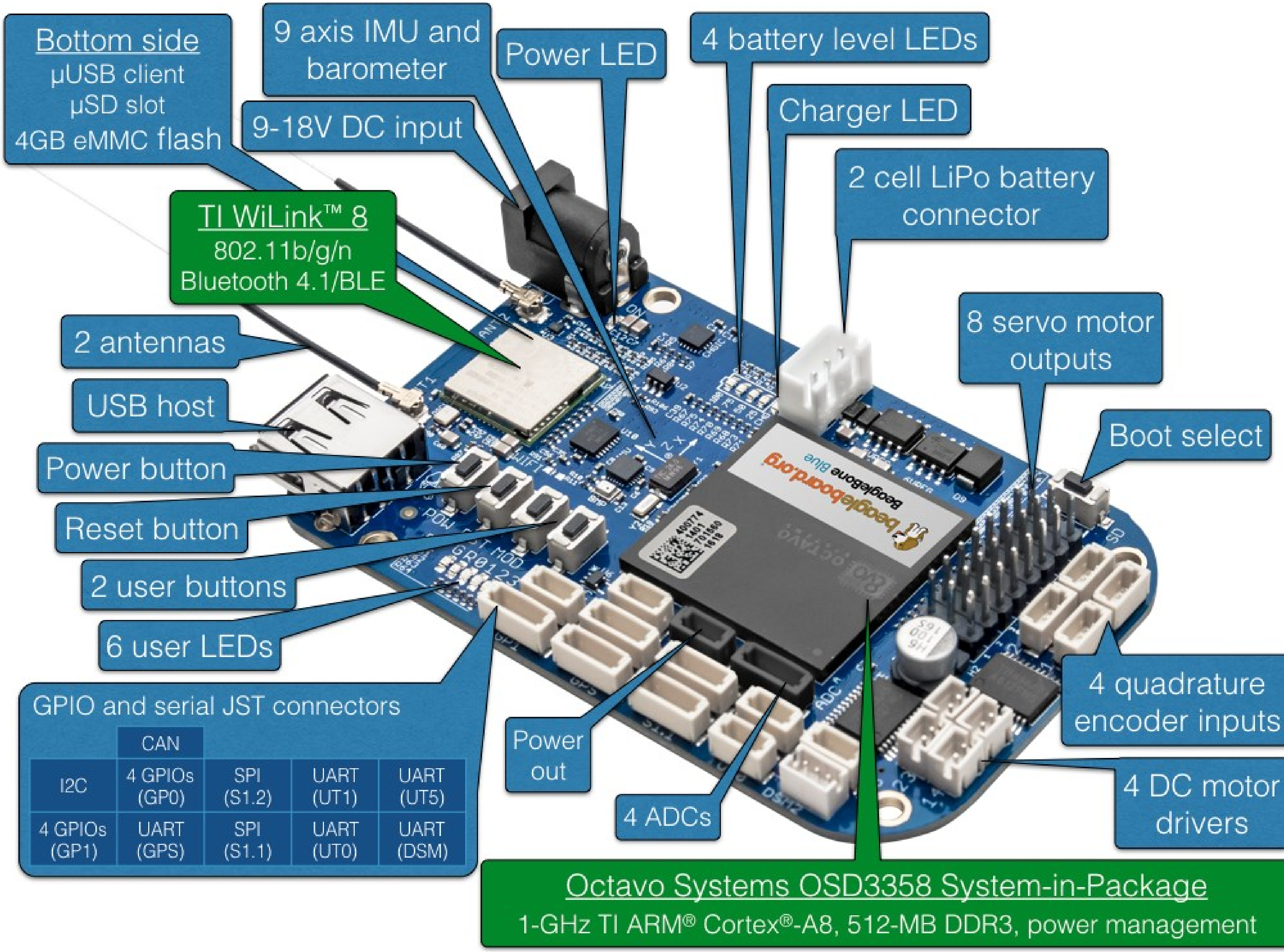


LINE CAMERA for EE192

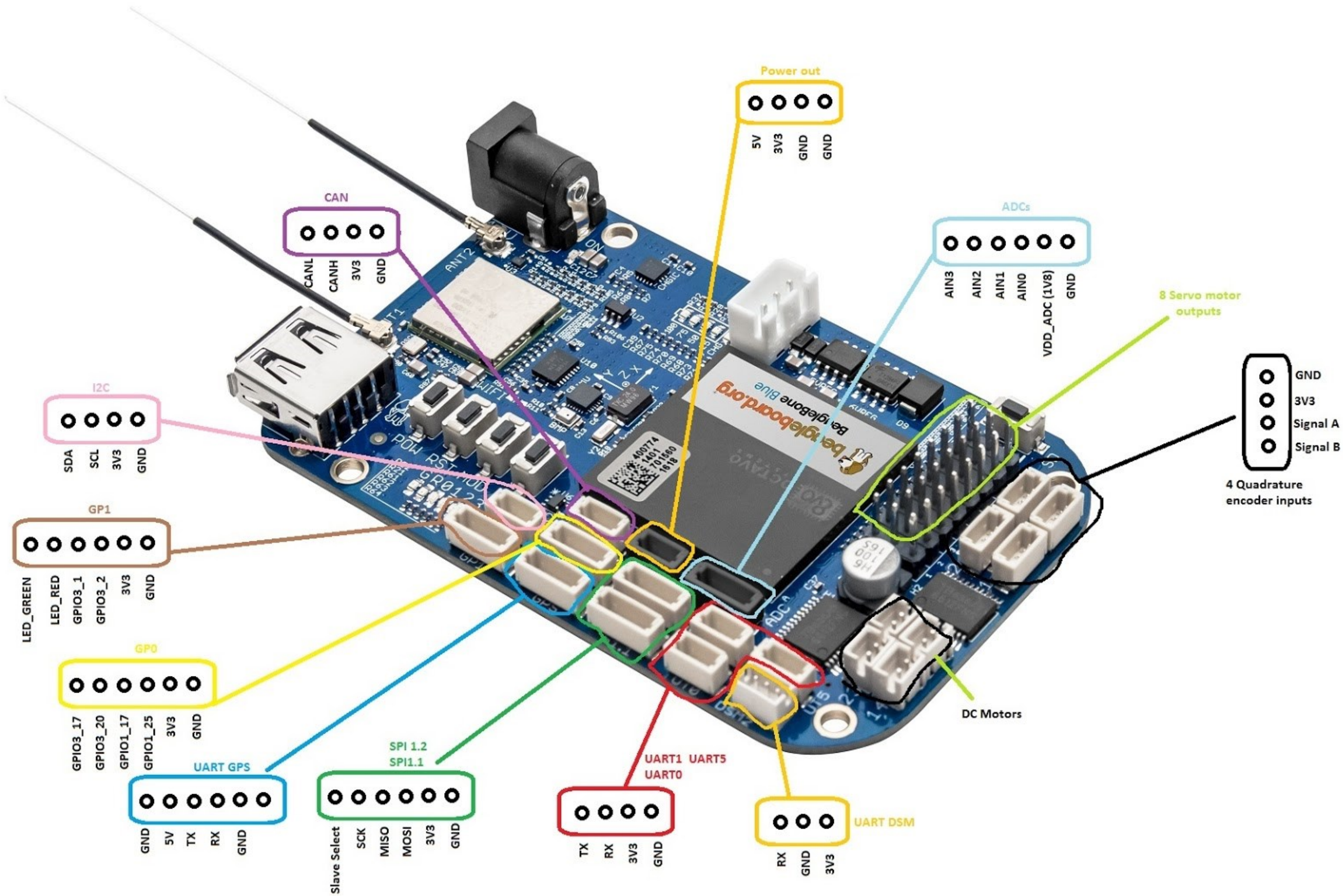
RC SERVO DRIVE

Used on BeagleBone Blue for real-time A/D for line camera and RC servo

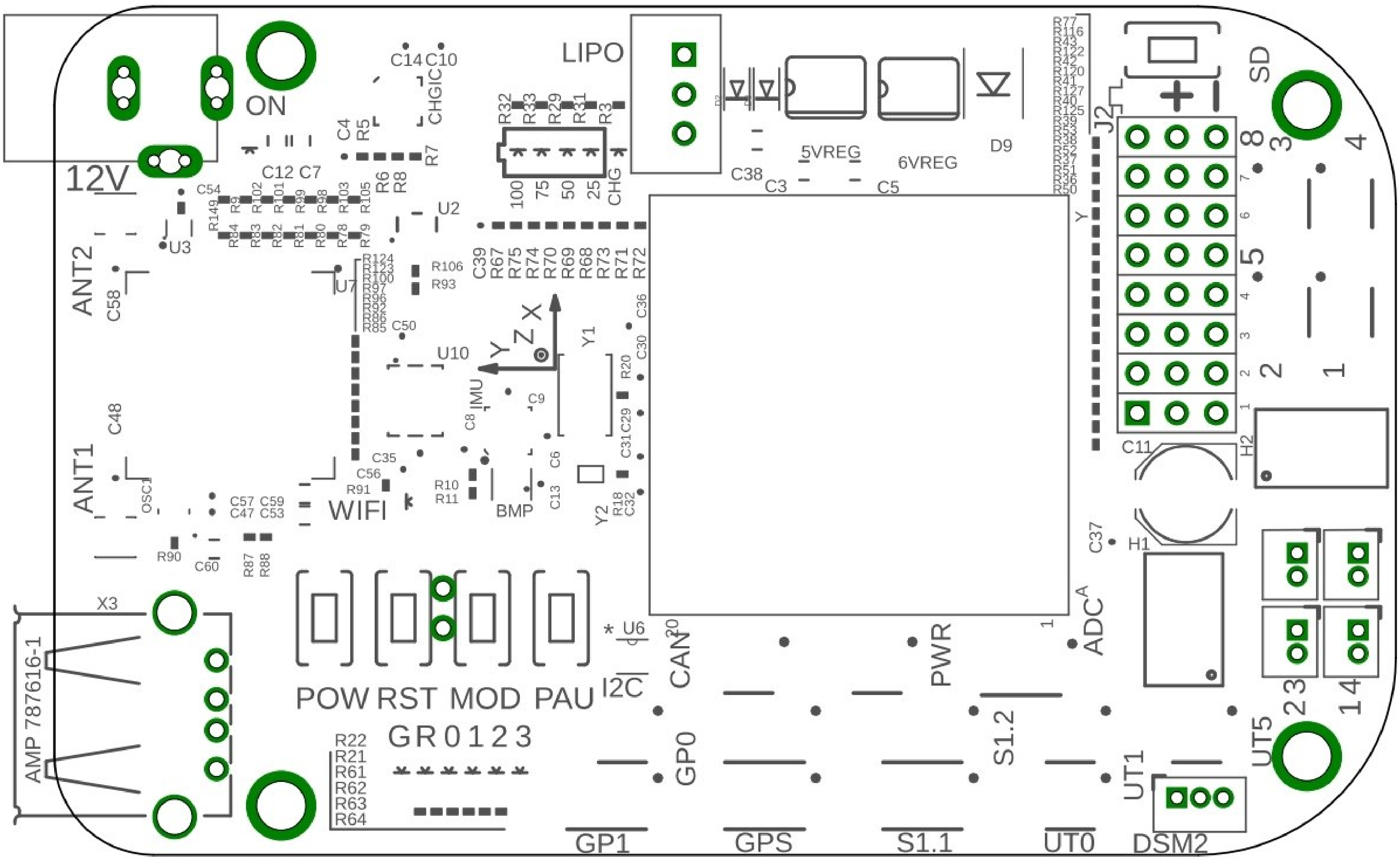
Beaglebone Blue Connections



Beaglebone Blue Connections



Beaglebone Blue Connections



Strawson Design librobotcontrol (1/2)

http://strawsondesign.com/docs/librobotcontrol/index.html

Algebra	Advanced linear algebra functions
SISO_Filter	Functions for generating and implementing discrete SISO filters
Kalman	Kalman filter implementation
Matrix	Functions for masic matrix manipulation
Other_Math	Math functions that don't fit elsewhere
Polynomial	Functions for polynomial manipulation
Quaternion	Functions for quaternion manipulation
Ring_Buffer	Ring buffer implementation for double-precision floats
Vector	Functions for vector manipulation

ADC	C interface for the Linux IIO ADC driver
GPIO	C interface for the Linux GPIO driver
I2C	C interface for the the Linux I2C driver
Pinmux	C interface for the Sitara pinmux helper driver
PWM	C interface for the Sitara PWM driver
SPI	General purpose C interface to the Linux SPI driver
UART	C interface for the Linux UART driver

Strawson Design librobotcontrol (2/2)

http://strawsondesign.com/docs/librobotcontrol/index.html

Quadrature_Encoder	Functions for reading quadrature encoders
Button	Handle generic GPIO buttons
CPU	Control CPU scaling governer
DSM	DSM2 and DSMX radio interface
LED	Control the LEDs on Robotics Cape and BeagleBone Blue
Model	Determine the model of board currently being used
Motor	Control 4 DC motor Channels
IMU_MPU	A userspace C interface for the invensense MPU6050, MPU6500, MPU9150, and MPU9250 - **GYRO**
PRU	Start and stop the PRU from userspace
Pthread	Manage pthreads and process niceness
Servo	Control Servos and Brushless Motor Controllers
Start_stop	Cleanly start and stop a process, signal handling, program flow
Time	Sleep and timing functions
Version	Macros and functions for getting the current version of librobotcontrol

Timing with Linux + PRU

librobotcontrol:
http://strawsondesign.com/docs/librobotcontrol/index.html

`POSIX threads: rc_pthread_create()`
(note random execution order, can tune priority)

// Returns the number of nanoseconds from when the calling thread was started in CPU time.

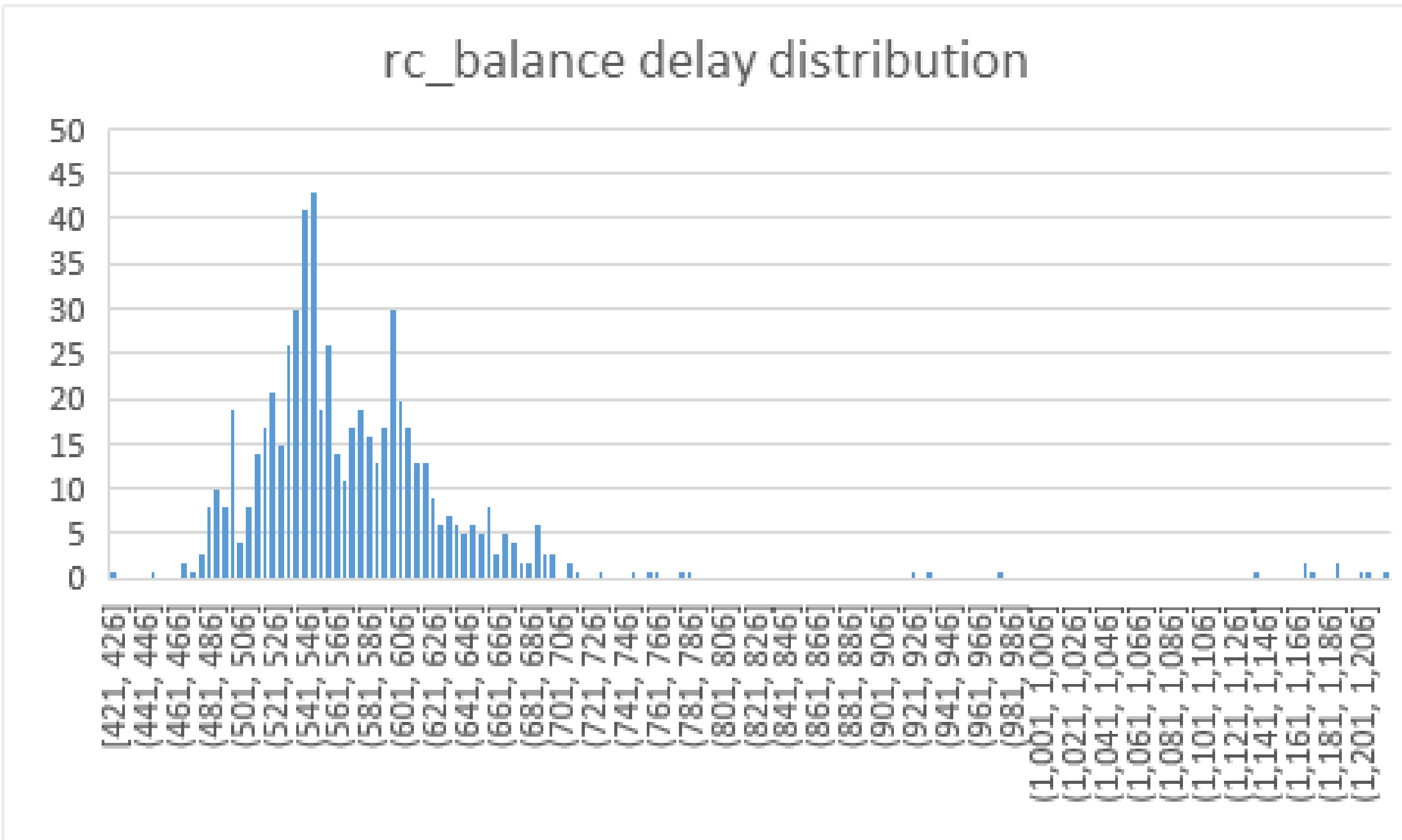
```
uint64_t rc_nanos_thread_time (   void  )
```

// Sleep in nanoseconds.

```
void rc_nanosleep (uint64_t  ns)
```

Example Timing Uncertainty in Linux: main control loop

Calculation loop: input/process/output. Note outliers.

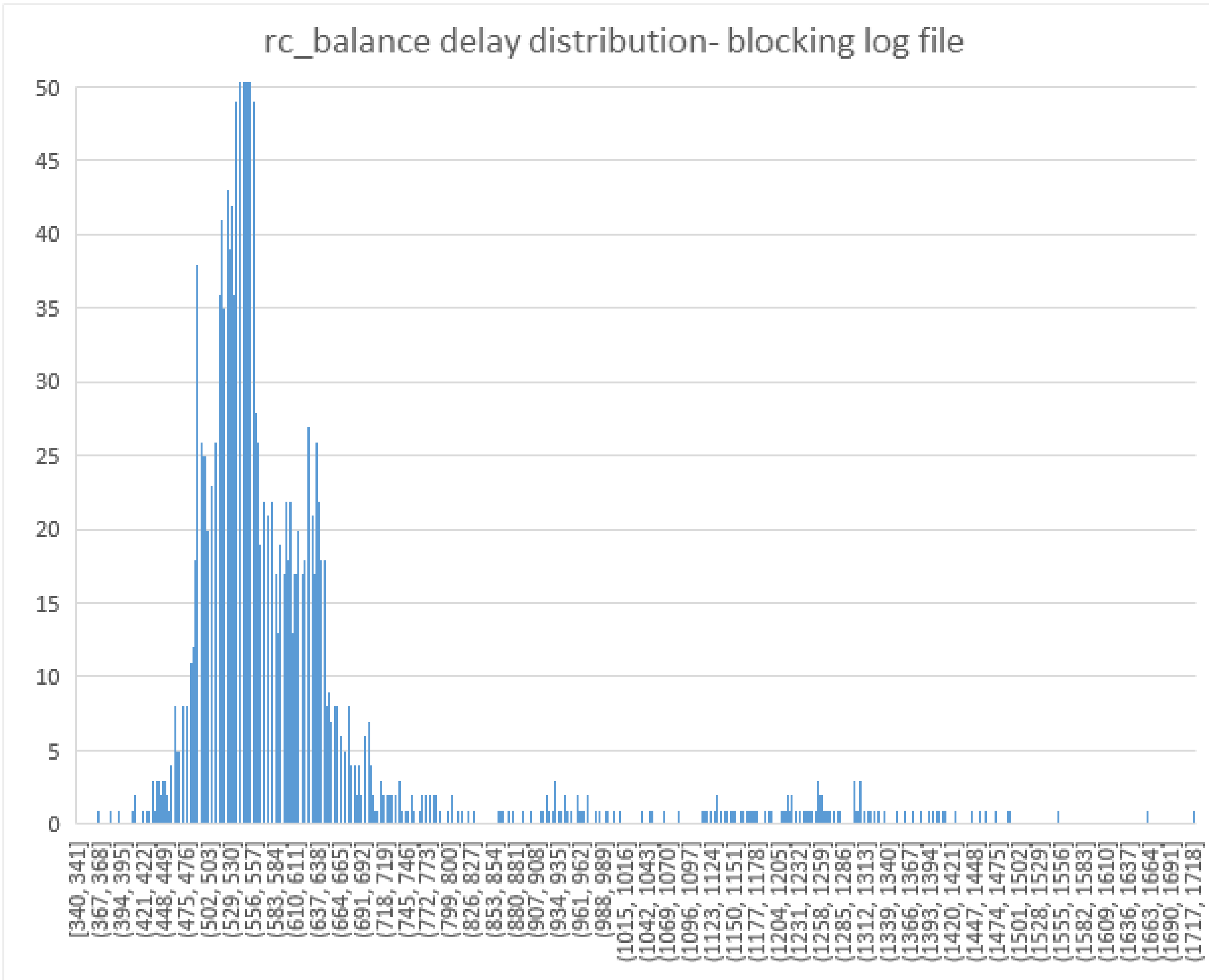


420 us

1200 us

Example Timing Uncertainty in Linux

Using printf in control loop (NOT RECOMMENDED)

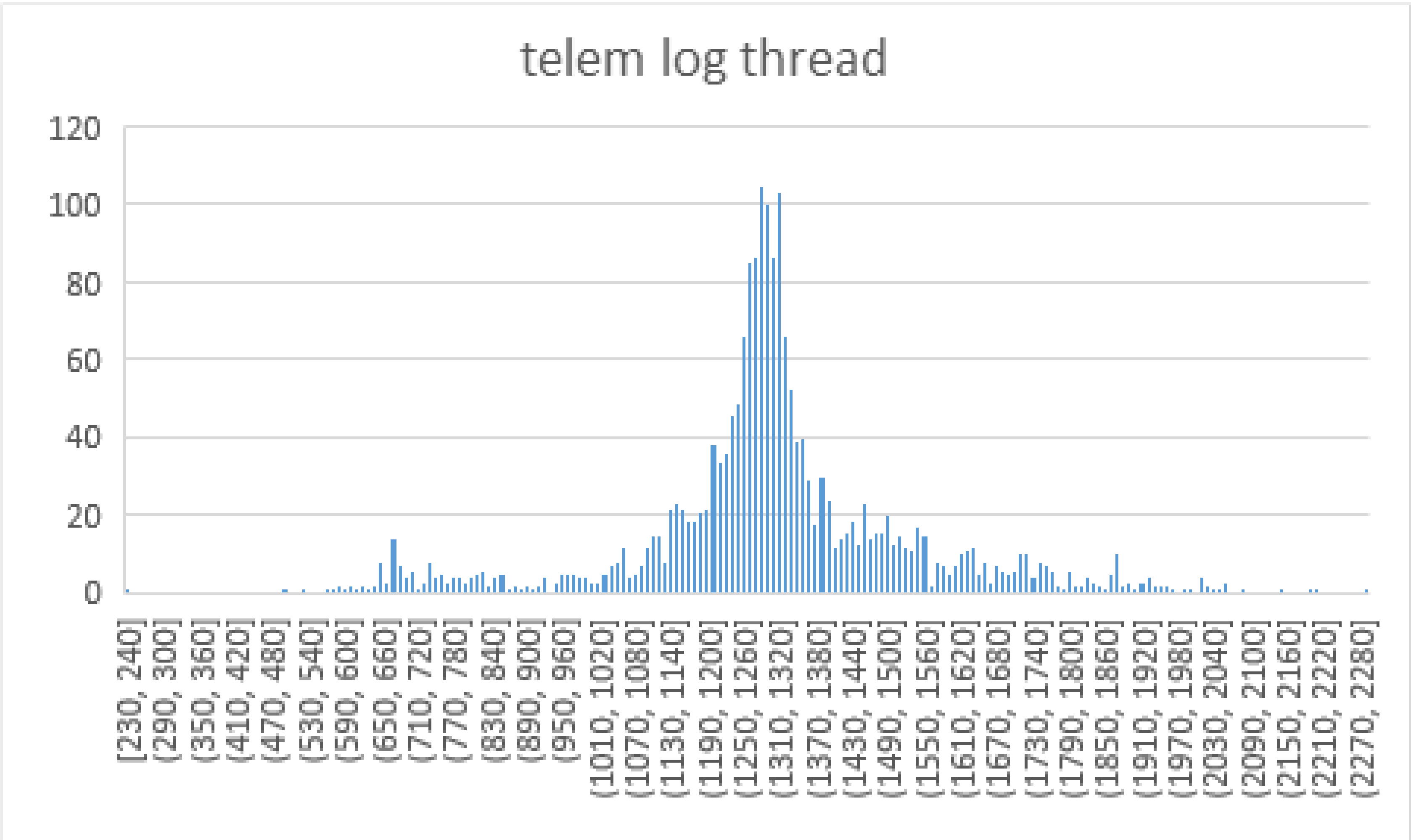


340 us

1700 us

Example Timing Uncertainty in Linux

Using fprintf in independent thread (better)



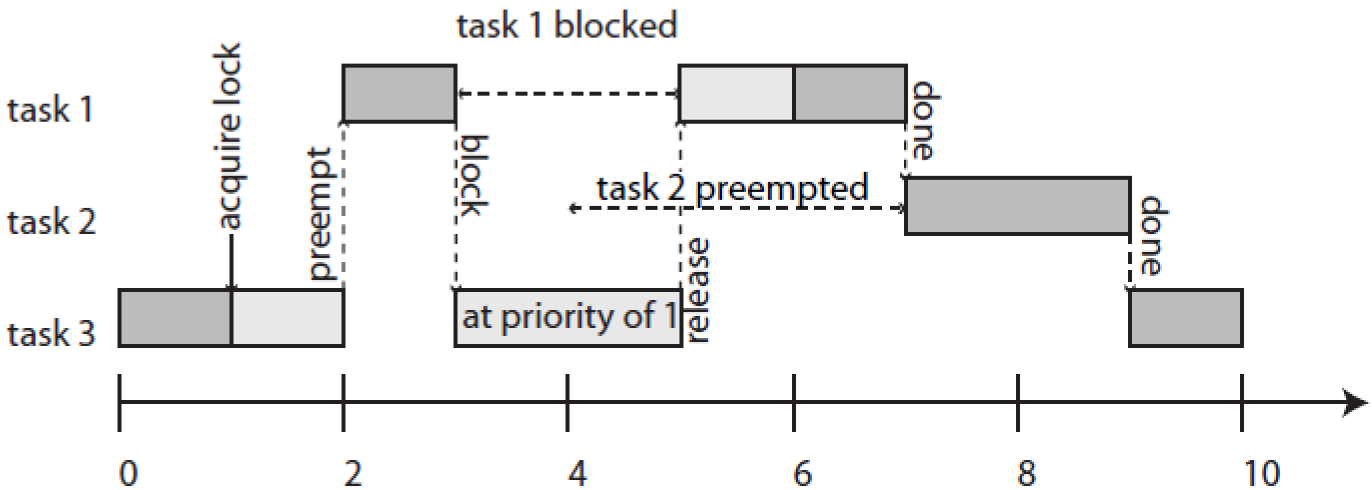


Figure 12.10: Illustration of the priority inheritance protocol. Task 1 has highest priority, task 3 lowest. Task 3 acquires a lock on a shared object, entering a critical section. It gets preempted by task 1, which then tries to acquire the lock and blocks. Task 3 inherits the priority of task 1, preventing preemption by task 2.

Can ROS Service be used in Real-time Application or Hardware Control?

https://answers.ros.org/question/266497/can-ros-service-be-used-in-real-time-application-or-hardware-control/

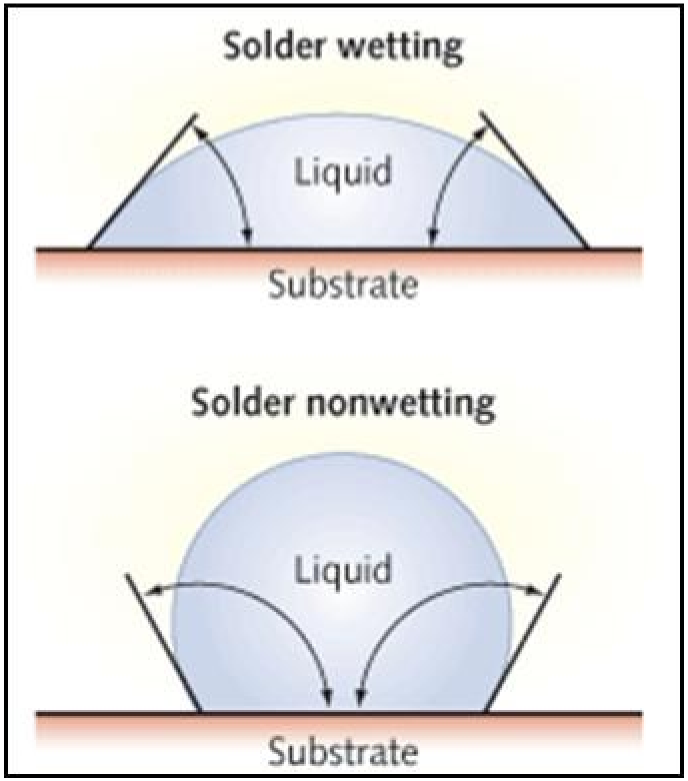
ROS does not provide any guarantees about request latency, scheduling delay, processing time or response latency when using services (or topics for that matter). For two nodes running on the same computer these delays are typically between a few milliseconds and a few hundred milliseconds.

Unlike topics, services are synchronous and the client must wait for the transport delay in both the request and the response.

The PR2 uses services, topics, non-blocking queues, and a few other real-time software techniques in the node that runs the real-time control loops.

The PR2 needs to run the control loop at 1kHz, so it also runs the real-time thread with increased priority on a linux kernel with the rt-preempt patches.

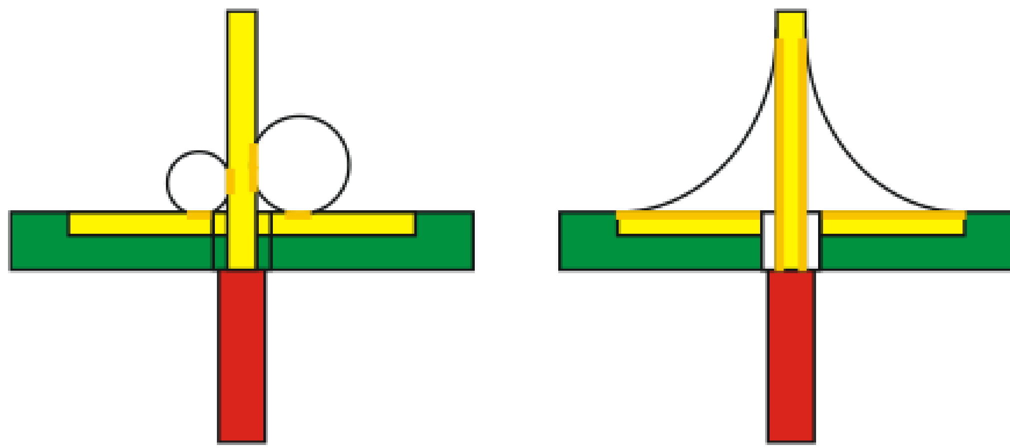
Electronic Interconnect and Components



From: http://solutions.palomartechnologies.com/Portals/60069/images//Wetting%20vs%20non-wetting%20conditions-resized-600.JPG

EE192- Soldering Notes

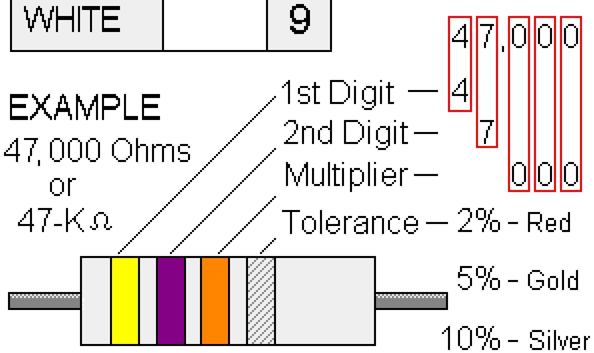
- Oxide has lower energy than clean metal
- Higher energy surfaces attract molten solder
- Oxides have higher melting points than metals
- Oxides have lower thermal conductivity than metals
- Flux helps to prevent oxide formation, but is an insulator



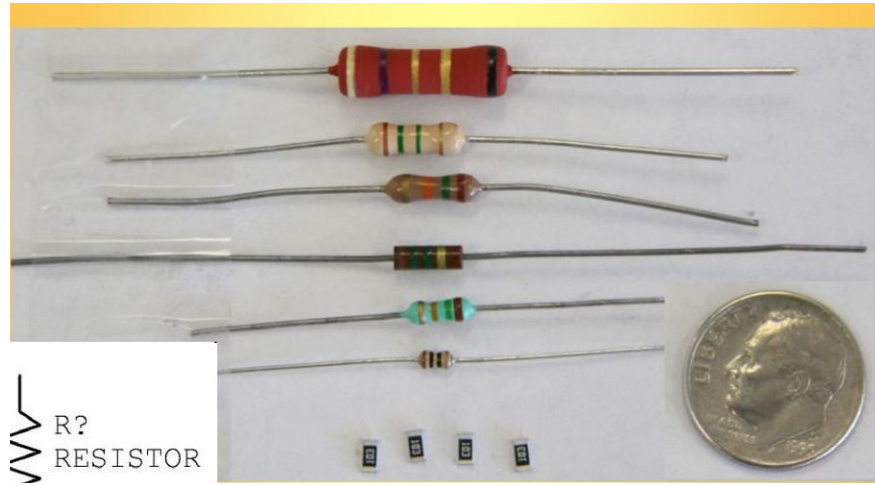
From:http://www.slagcoin.com/joystick/pcb_wiring/bond.png

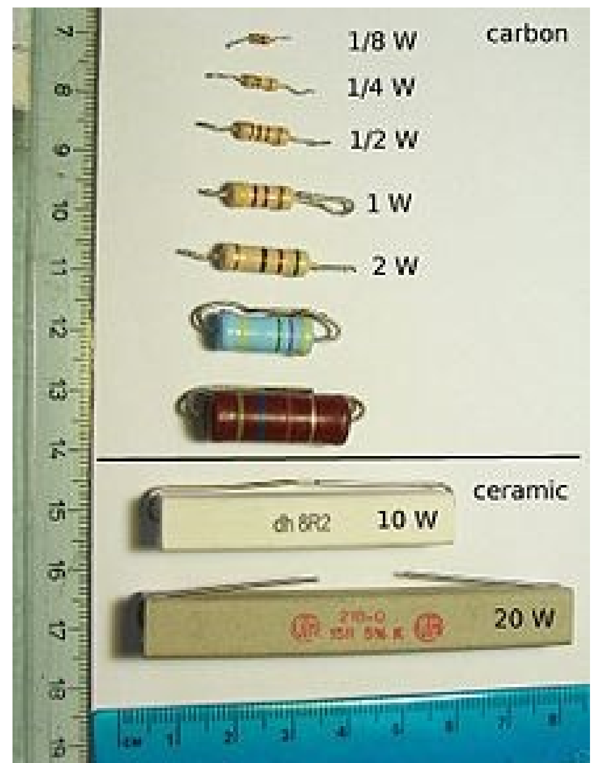
Electronic Components- Resistors

Color		Digit	Multiplier
BLACK		0	______
BROWN		1	_____0
RED		2	____00
ORANGE		3	___000
YELLOW		4	__0, 000
GREEN		5	_00, 000
BLUE		6	000, 000
VIOLET		7	
GRAY		8	
WHITE		9	



Yellow | violet | orange| gold

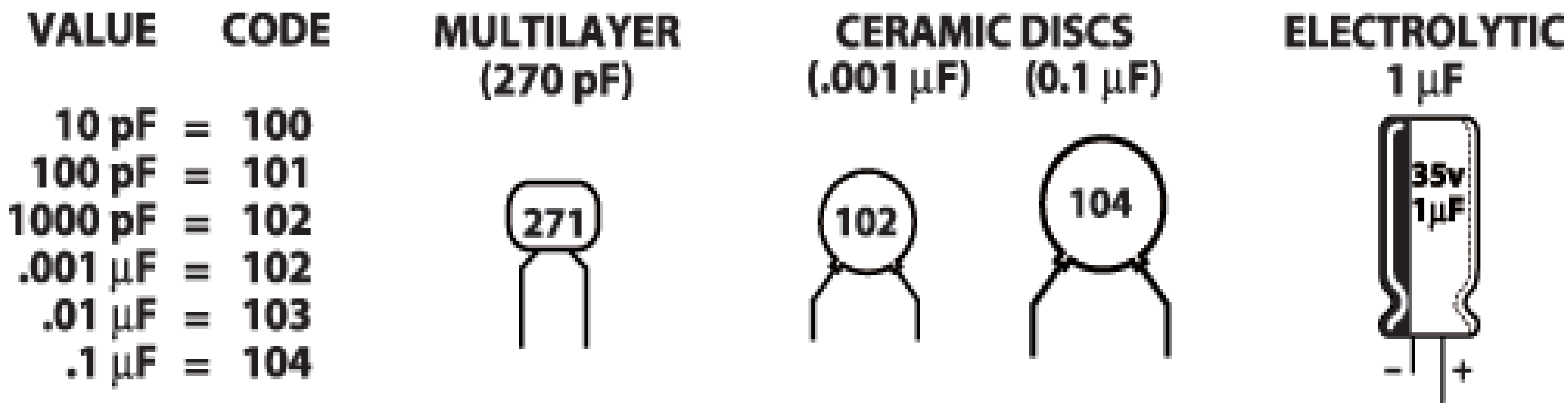




Better be right or your great big venture goes west…

Capacitor Codes

From:http://www.applefritter.com/sites/default/meta/replicacreation/images/fige-10.png

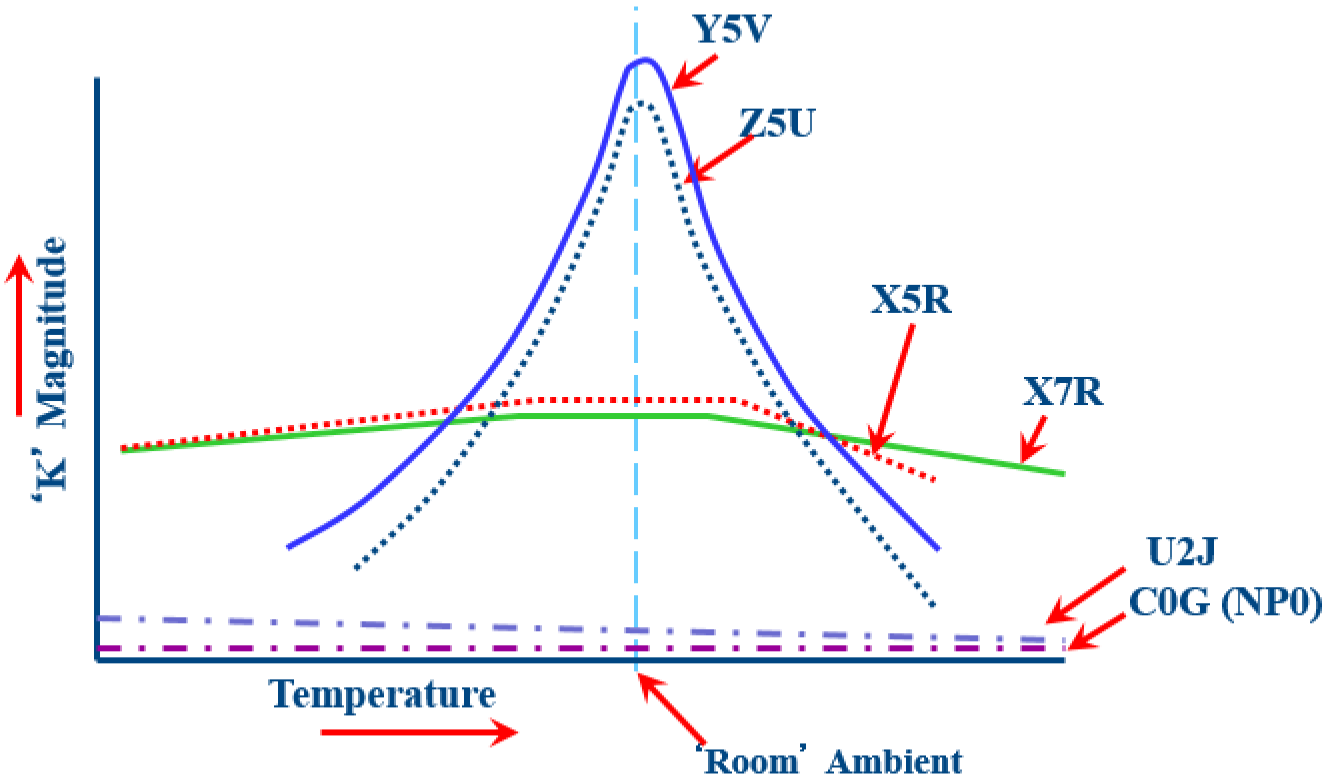


Capacitor Types-ceramic

CAP CER 0.1UF 50V X7R
RADIAL

0.1µF ±20% 50V Ceramic
Capacitor Z5U Radial

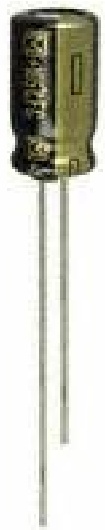
CAP CER 0.1UF 630V X7R RADIAL



https://ec.kemet.com/wp-content/uploads/2015/12/ceramic-dielectric-comparison-chart.png

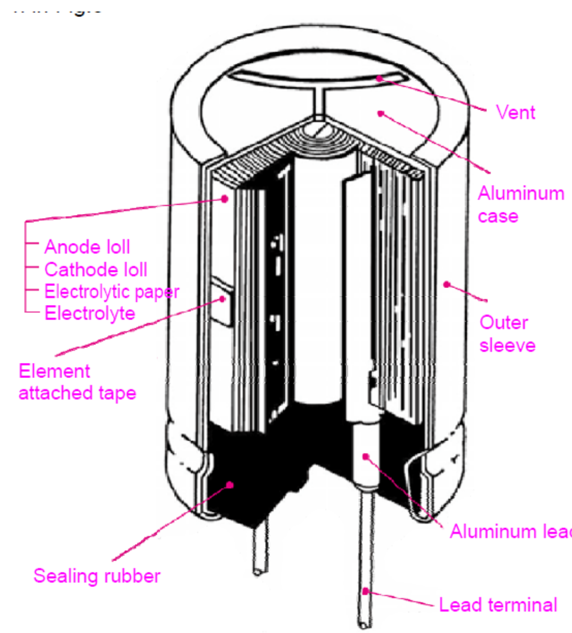
CAP CER 0.1UF 50V X7R 0805

Capacitor Types- 47 uF 50V

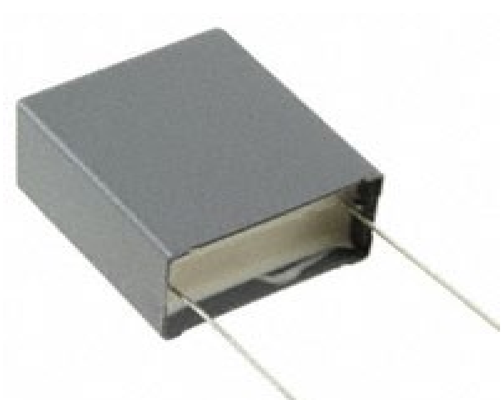


Ripple Current
600mA

Eletrolytic Ripple Current
169mA @ 120Hz



https://industrial.panasonic.com



Metalized film

CAP TANT 22UF 50V
20% 2917

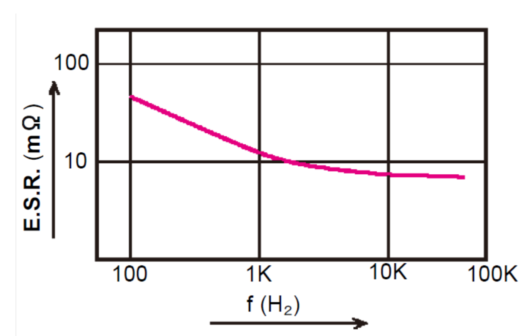


Fig.13 ESR vs frequency