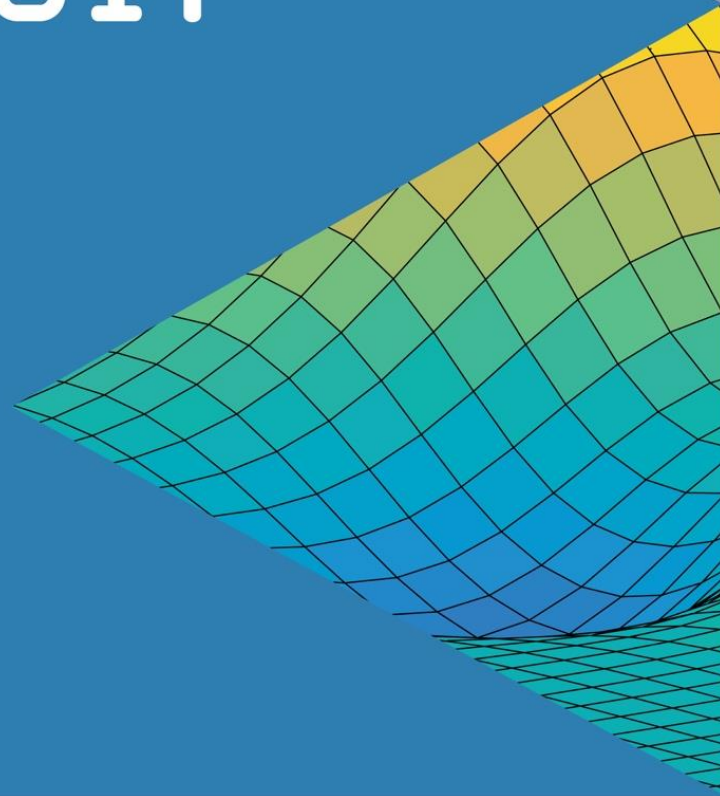


MATLAB EXPO 2017

Déploiement embarqué et
connectivité hardware avec
MATLAB et Simulink

Paul Cox, MathWorks



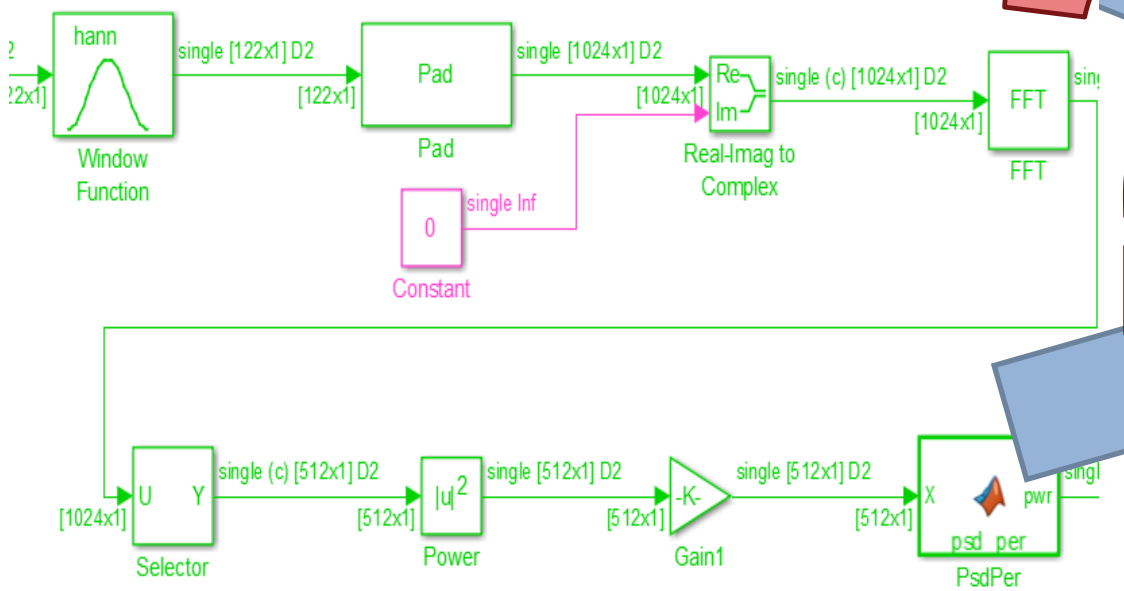
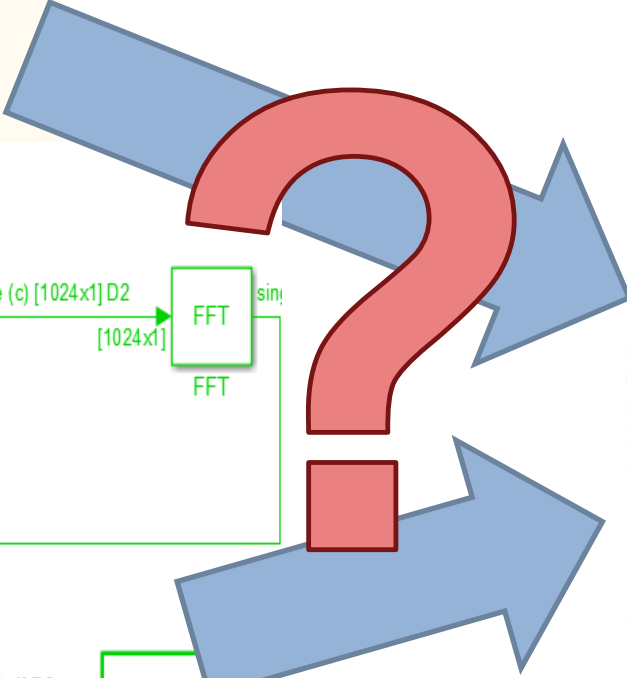
Agenda

- **Introduction**
- Hardware Support Packages for MATLAB and Simulink
- Processor-in-the-Loop Execution
- Code Generation within the Internet of Things (IoT)
- Conclusion
- Questions

From algorithm to hardware: why and how?

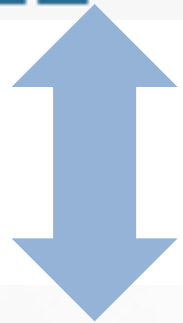
```

U = WIN.^WIN; % SCALAR, 1
% FFT
X = x.*win; % n-by-1
X = fft(x,nfft); % complex
X = X(1:nfft/2+1); % (nfft/2+1)
Pxx = X.*conj(X)/U;
Pxx = 2*Pxx/fs; Pxx(1)
df = fs/nfft;
    
```

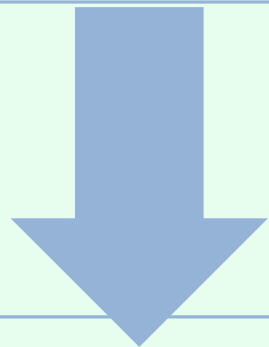


From MATLAB and Simulink to Hardware

Automatic Code Generation

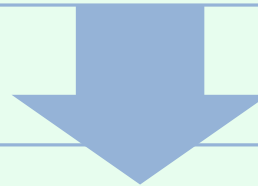


C/C++



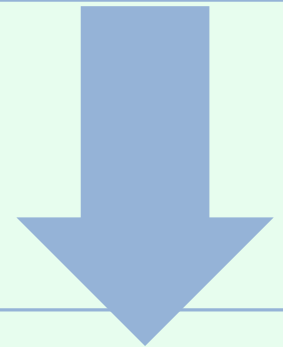
CPU
DSP

VHDL
Verilog



ASIC
FPGA

ST



PLC

Code Generation in Industry

ABB Accelerates the Delivery of Large-Scale, Grid-Connected Inverter Products with Model-Based Design

“Simulink and Embedded Coder enabled us to open the door to new markets. With increased productivity from extensive simulation and efficient code generation, we have confidence in our ability to produce the systems that larger customers are asking for in the time frames they want.”

— Dr. Robert Turner, ABB



Code Generation in Academia

NASA Interns Develop Guidance, Navigation, and Control Software for Quadcopter with Model-Based Design

Model-Based Design makes both working engineers and interns at NASA MSFC more productive. The students have more fun because they can run the GNC algorithms they create in Simulink on a real processor and quickly get things done.

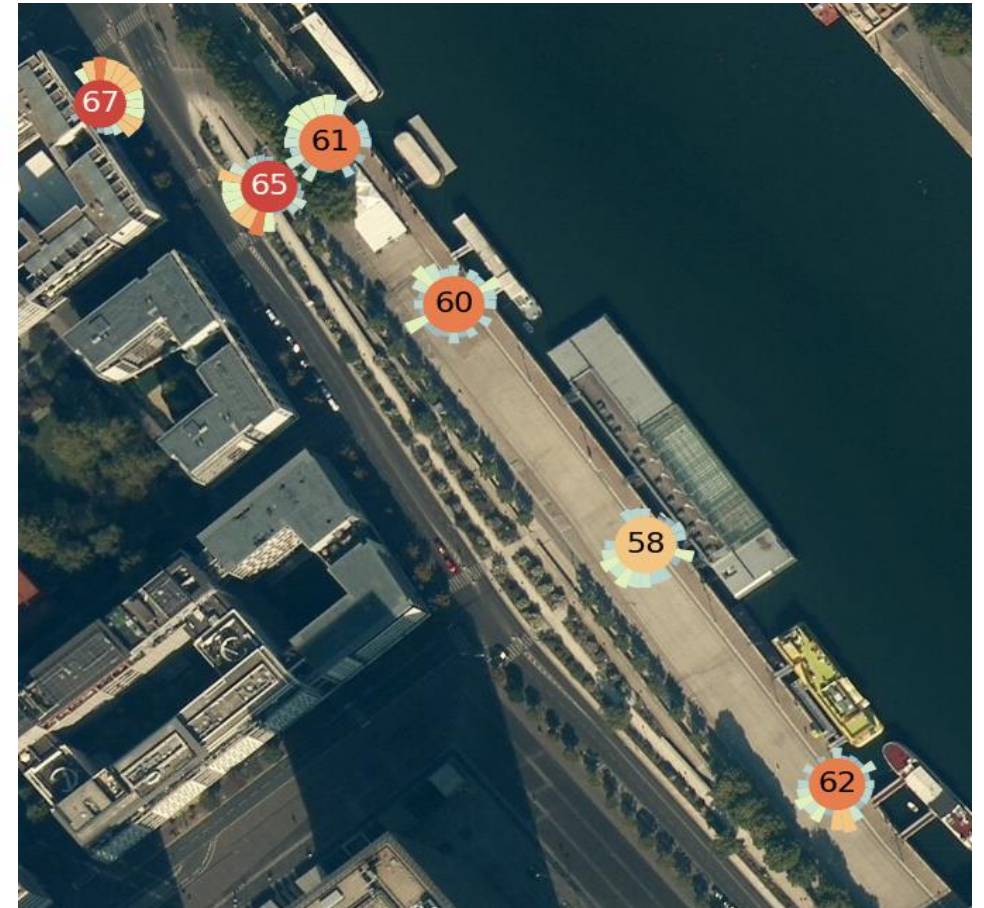


NASA intern working with the quadcopter vehicle and ArduPilot Mega 2.5 hardware.



Code Generation Case Study : Bruitparif Medusa

Noise monitoring distributed network



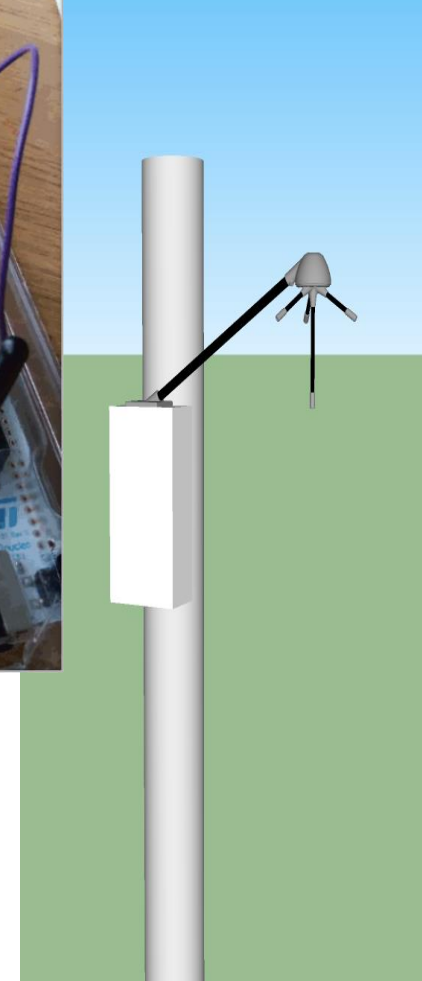
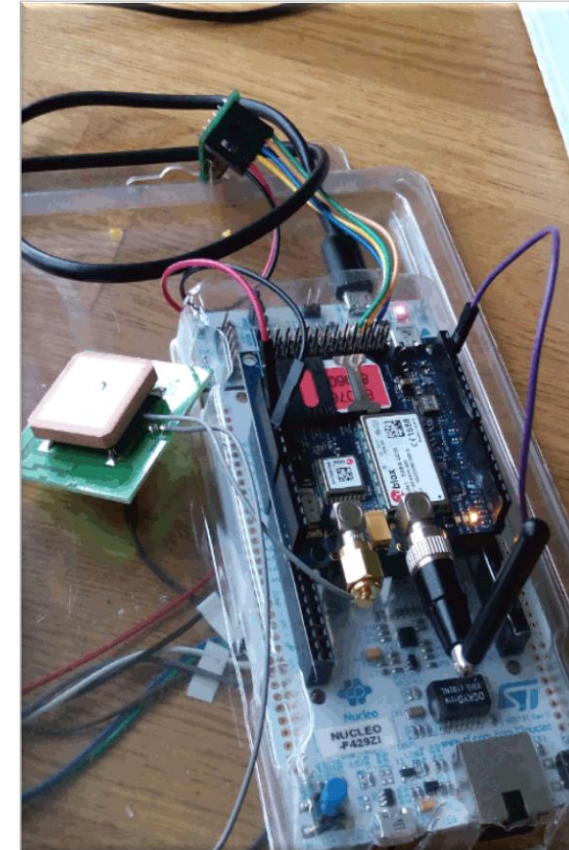
Code Generation Case Study : Bruitparif Medusa

- Requirements

- Monitor noise levels and directions
- Send live data to a central server
- Low-power consumption device ($< 5\text{ W}$)
- Minimal network traffic ($< 5\text{ kbps}$)

- Solution

- Low-power microcontroller (STM32F4)
- UDP communication over cellular network
- Algorithm development with MATLAB/Simulink
 - Design and feasibility/performance analysis
 - Preparation for embedded deployment
 - Automatic Production Code Generation with Embedded Coder
 - Low memory and CPU utilization

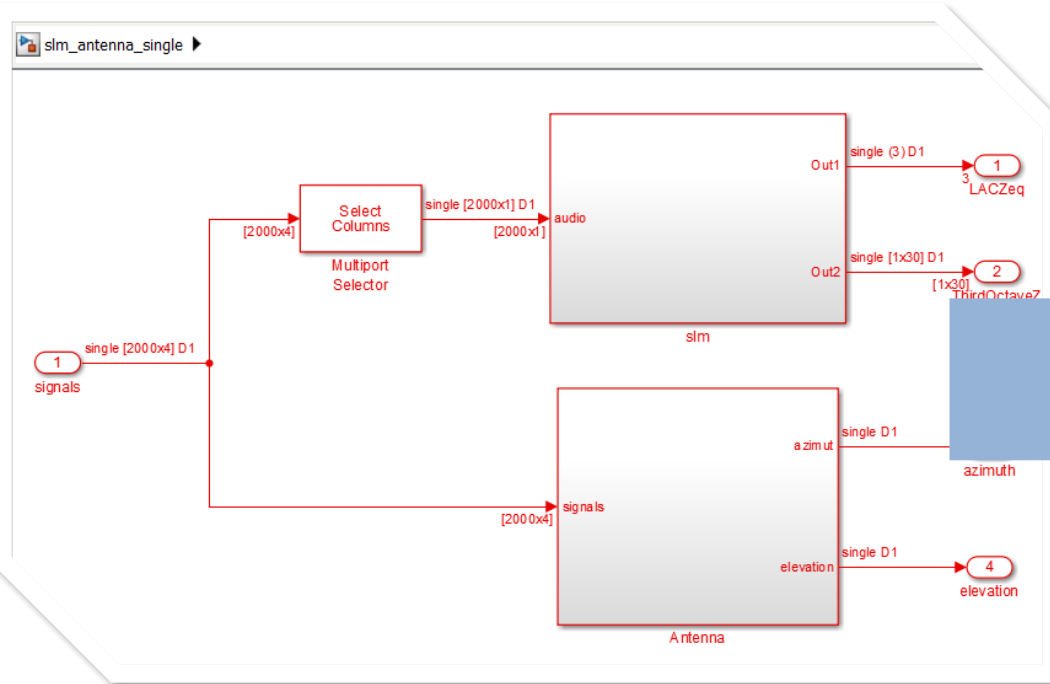


Bruitparif - Software

Total time (seconds × 1e-09)	1583320256
Measured time display options	('Units', 'Seconds', 'ScaleFactor', '1e-09', 'NumericFormat')
Timer frequency (ticks per second)	1.68e+08
Profiling data created	19-Mar-2016 18:38:53

2. Profiled Sections of Code

Section	Maximum Execution Time	Average Execution Time	Maximum Self Time	Average Self Time	Calls
bot_initialize	1060	1060	1060	1060	1
[+] bot_Init	47048	47048	32869	32869	1
[+] bot [0.015625 0]	5405875	5373712	1575119	1568924	129
Select mic	42530	42530	42530	42530	129
Generated Filter Block	85125	84842	85125	84842	129
Generated Filter Block	57113	57105	57113	57105	129
Math Function	26042	23352	26042	23352	129
Math Function1	26161	23760	26161	23760	129



1. Function replacements in bot [hide]

The following table provides a mapping from the functions used from the selected Code Rep blocks in the model that triggered the replacement.

Function	Block
arm_biquad_cascade_df2T_f32	<S3>/Generated Filter Block <S4>/Generated Filter Block
arm_cfft_radix2_f32	<S8>/IFFT <S9>/IFFT

Bruitparif – Software (continued)

The screenshot displays the MATLAB IDE interface. On the left, the 'Functions' pane shows a tree view of files, with 'antenna_gphat_calib_detect.c' and its functions 'antenna_slm_init (void)' and 'antenna_slm_step (real32_T arg)' highlighted with a red box. The main editor window shows the source code for 'antenna_gphat_calib_detect.c'. The code includes a loop for reading microphone data from four channels (MIC 1 to MIC 4) and normalizing the values. A red box highlights the function call 'antenna_slm_step(arg_mics, &arg_azimut, &arg_elevation, &arg_valid, arg_n)' at line 759. The 'Build Output' pane at the bottom shows the compilation progress for the target 'medusa_f429_rtos'.

```

738 //HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
739 if(osSemaphoreWait(dmaSaiBCpltSemHandle, osWaitForever) == osOK) {
740 HAL_GPIO_TogglePin(LD3_GPIO_Port, LD2_Pin);
741 HAL_WWDG_Refresh(&hwwdg, WDOG_TO);
742 for (int i = 0; i < NB_SAMPLES; i++){
743     v_mic1 = *(int32_t*)(bufferSaiA + 8 * (i + NB_SAMPLES)); // MIC 1
744     v_mic2 = *(int32_t*)(bufferSaiA + 4 + 8 * (i + NB_SAMPLES)); // MIC 2
745     v_mic3 = *(int32_t*)(bufferSaiB + 8 * (i + NB_SAMPLES)); // MIC 3
746     v_mic4 = *(int32_t*)(bufferSaiB + 4 + 8 * (i + NB_SAMPLES)); // MIC 4
747
748     v_mic1 = v_mic1 / 256;
749     v_mic2 = v_mic2 / 256;
750     v_mic3 = v_mic3 / 256;
751     v_mic4 = v_mic4 / 256;
752
753     arg_mics[i] = (real32_T)v_mic1 / 8388608;
754     arg_mics[250 + i] = (real32_T)v_mic2 / 8388608;
755     arg_mics[500 + i] = (real32_T)v_mic3 / 8388608;
756     arg_mics[750 + i] = (real32_T)v_mic4 / 8388608;
757 }
758
759 // Compute Levels & Angles
760 antenna_slm_step(arg_mics, &arg_azimut, &arg_elevation, &arg_valid, arg_n);
761
762 // Increment counter
763 cntSteps += 1;
764
765 //len = sprintf(msg2, "%d\r\n", cntSteps);
766 //HAL_UART_Transmit(&uart3, (uint8_t*)msg2, len, 0xFFFF);
767
768 if (cntSteps == nbIter) {
769
770     // if (!arg_calibFlag[0] && !arg_calibFlag[1] && !arg_calibFlag[2])
771         HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);

```

Build Output

```

Build target 'medusa_f429_rtos'
compiling main.c...
linking...
Program Size: Code=75296 RO-data=40128 RW-data=62548 ZI-data=198636
"medusa_f429_rtos\medusa_f429_rtos.axf" - 0 Error(s), 0 Warning(s).

```

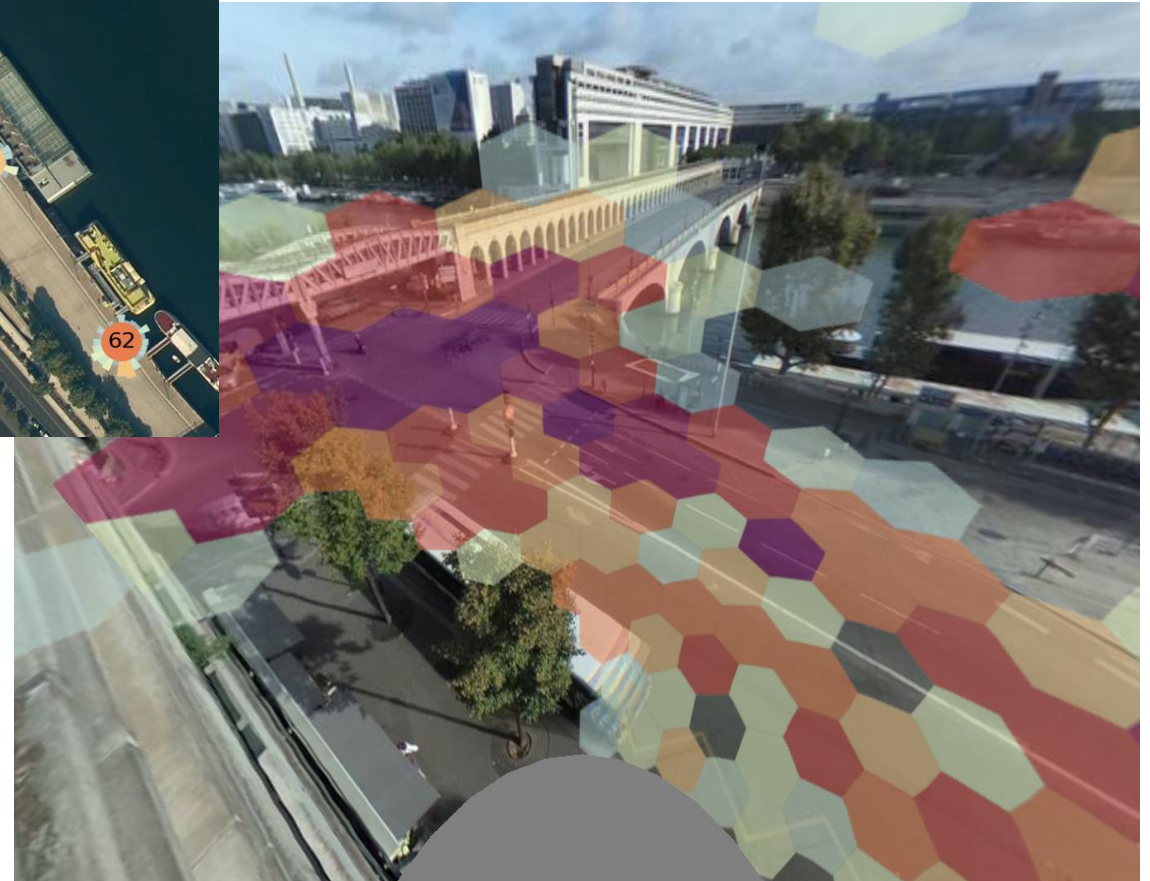
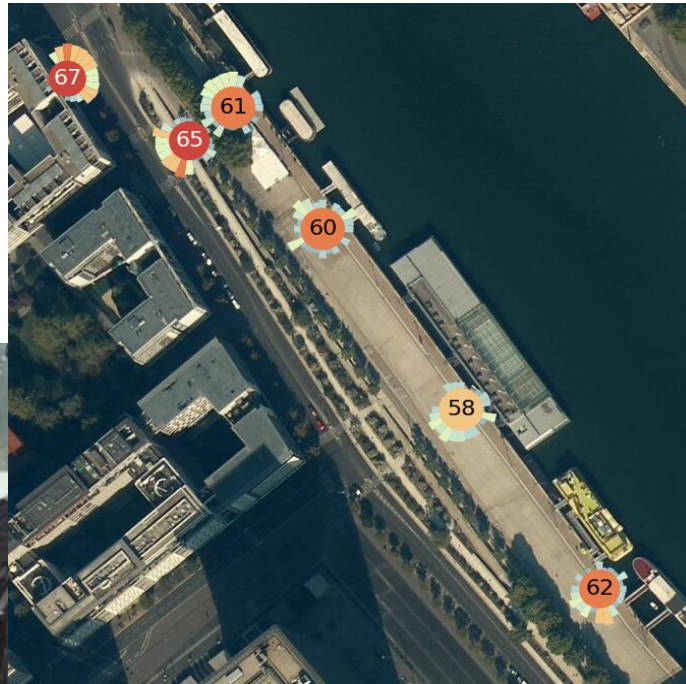
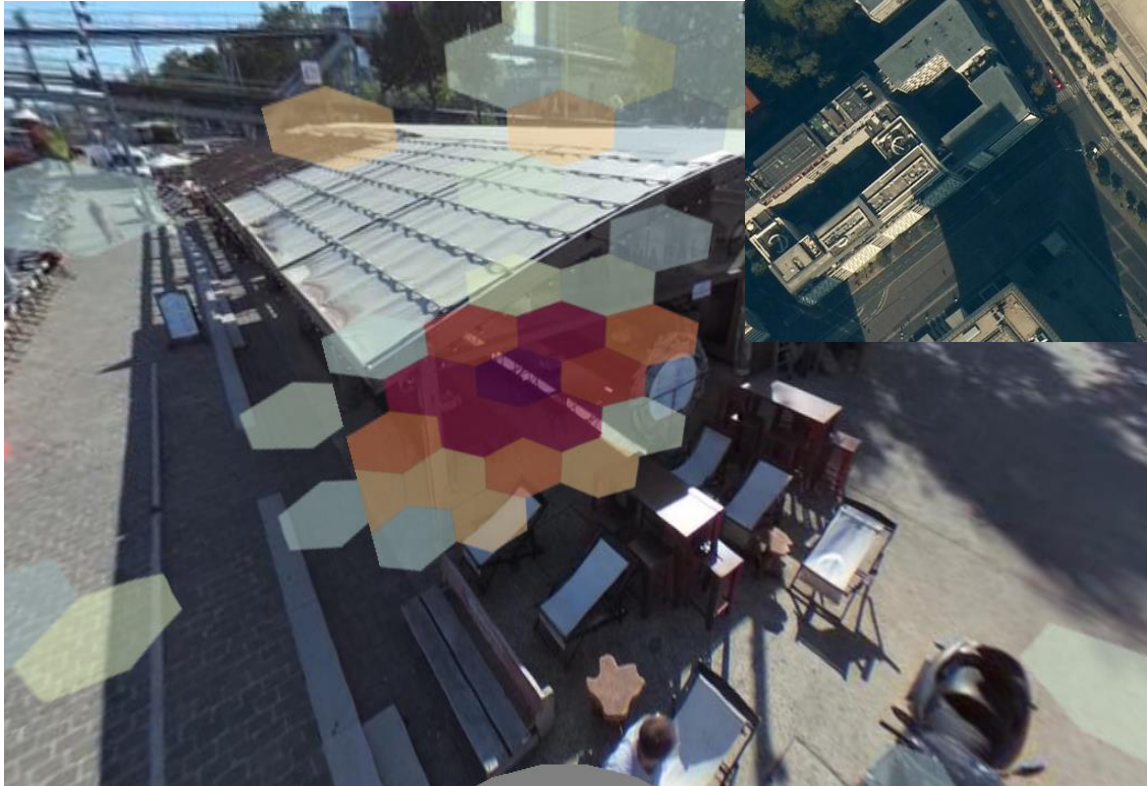
Automatic code generation in IoT network nodes



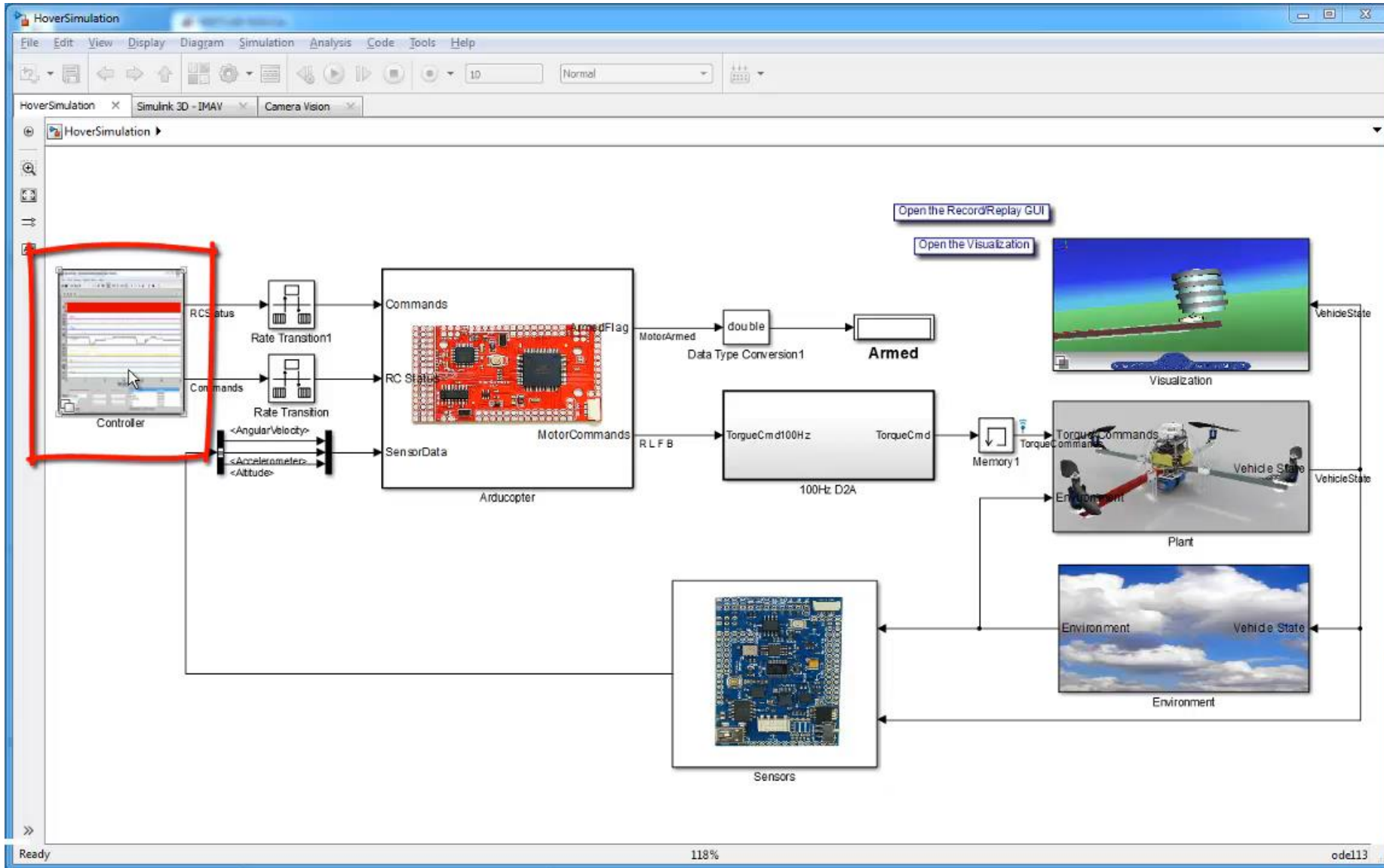
Observatoire du bruit



Bruitparif – Pilot Project Results



Before Code Generation: Modeling and Simulation



Code Generation for UAV Team MAVerix at student competition

The image shows two windows from the Simulink environment. The top window is titled "FluCo Linker Script" and displays a list of parameters and their values, such as "SEKND Ch.1-7", "SEKND Ch.1-11", "PMD", "CND", "ACN", "ACTN", "POT1", "SPF_HWT", and "SPF_HSL". The bottom window is titled "FluCo Parameter & Scopes" and shows a block diagram with a central green block and several red blocks connected to it.

The image shows a "Command Window" with the title "FluCo Code". It displays the following text:

```
## START
## User Request
C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe
## ADVERTISED MODEL
C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe
## Starting build procedure for model: Regler
## Generating code into build folder: C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe
## Invoking Target Language Compiler on Regler.c
## Using System Target File: C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe C:\Program Files\MathWorks\Simulink\bin\win32\slsh.exe
## Loading ILC Runtime Libraries
*****
## Initial gain through model to make user defined code
*****
## Getting model source code
## Writing header file Regler.h
## Writing source file Regler.c
## Writing source file evtdrv.c
## Writing header file rt_stylesheet.h
## Writing source file rt_stylesheet.c
## Writing
```



Awarded at IMAV 2013 and IMAV 2014 !



2016: 8 km DHL flight tests !



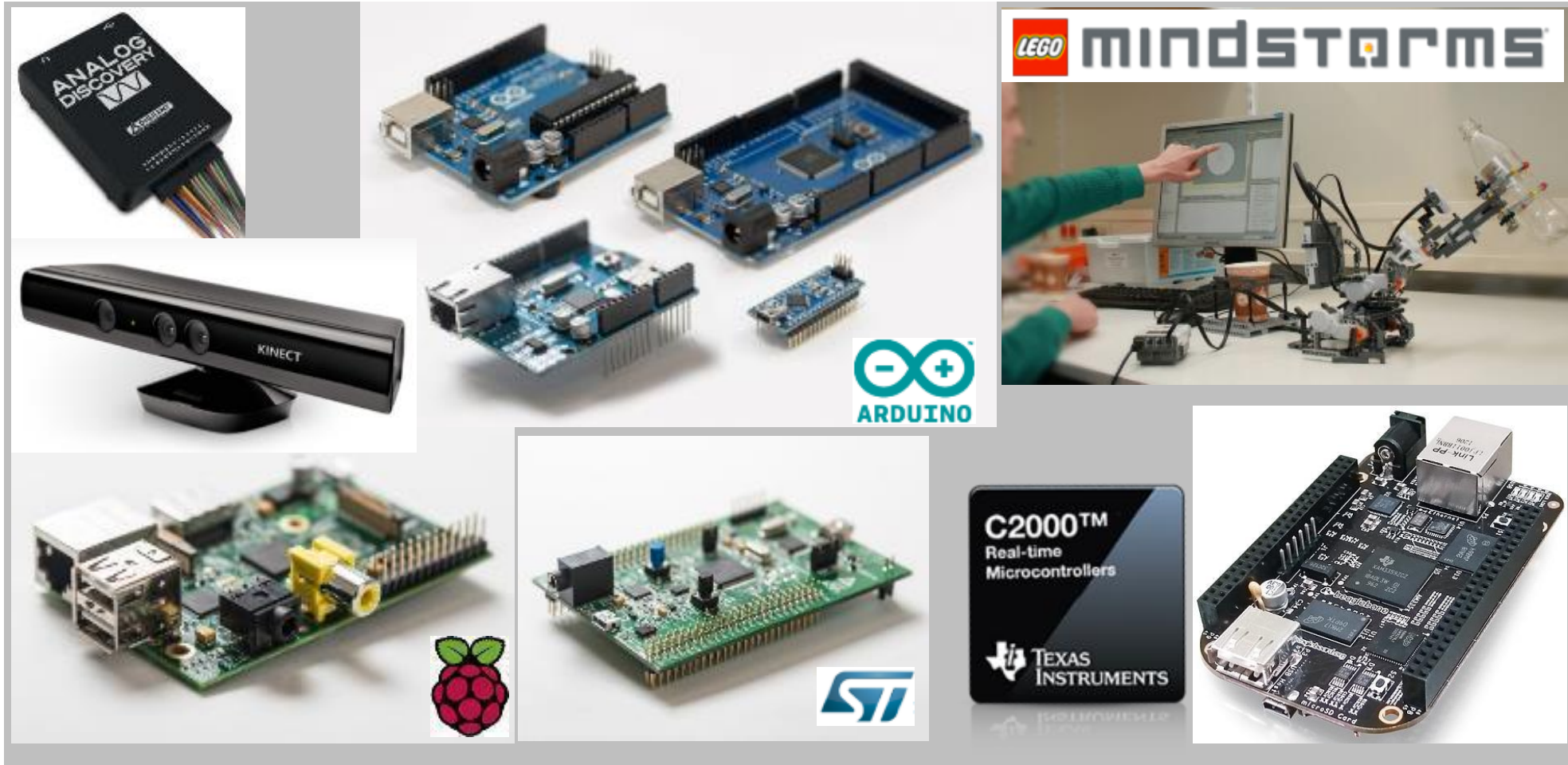
http://www.dpdhl.com/en/media_relations/specials/parcelcopter.html

Agenda

- Introduction
- **Hardware Support Packages for MATLAB and Simulink**
- Processor-in-the-Loop Execution
- Code Generation within the Internet of Things (IoT)
- Conclusion
- Questions

How do I connect MATLAB to hardware?

Since 2012: Hardware Support Packages!



C Code Generation-based Hardware Support Packages

- Texas Instruments C2000 *200+ user installs / month!*
- STmicroelectronics STM32F407 and STM32F746 Discovery boards
- Beaglebone Black
- Raspberry Pi 1, 2, 3 *500+ user installs / month!*
- Arduino (Uno, Due, Nano, Mini, Mega, too many to list!) *2000+ user installs / month!*
- NXP FRDM, STM32 Nucleo boards
- Android, iOS, and **more!** *250+ user installs / month!*

Includes: **Simulink-based HSP**

- Compiler Toolchain
- Peripheral configuration I/O Blocks
- External mode
- Processor-In-the-Loop PIL framework
- Example models and documentation

Includes: **MATLAB-based HSP**

- Connectivity API
- Coming Soon: Code Generation

Open Hardware Revolution



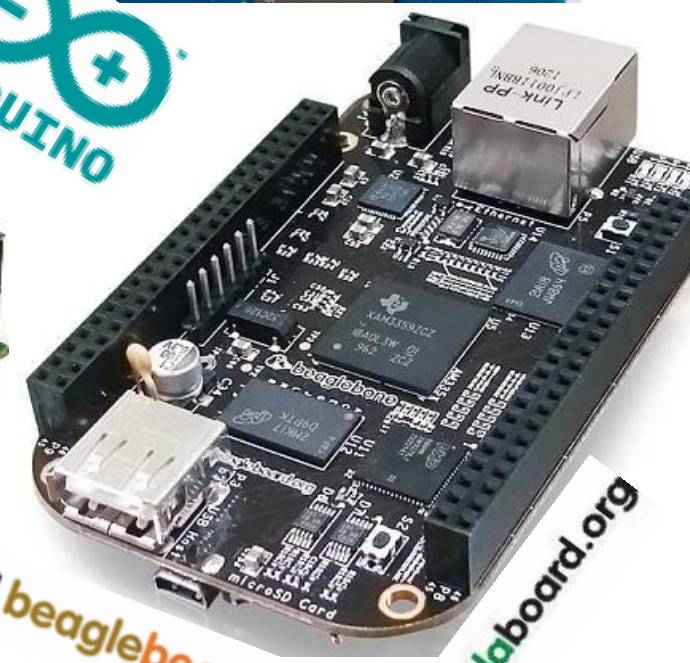
Execute



ANDROID



Deploy to Hardware



Parrot AR Drone

Execute



Parrot AR Drone Hardware Support in Simulink



Simulink Library Browser

File Edit View Help

Enter sea...

Libraries

- Simulink
- AR Drone 2 Target**
- Aerospace Blockset
- Communications Sys
- Computer Vision Sys
- Control System Tool
- DSP System Toolbox
- Embedded Coder
- Embedded Coder Su

Library: AR Drone 2 Target Search Results: (none)

Battery Voltage Measurement	Battery Volt- age Measur...	Inertial Measurement
Init Actuator	Init_Actuator	LED
Motor	Motor	Navigation Subsystem



Deploying Simulink Model to Parrot AR Drone

The screenshot displays the MATLAB/Simulink environment. On the left, the 'Euler Angles' scope shows three plots for Roll, Pitch, and Yaw over a 140-second period. The Roll plot ranges from -60 to 40, Pitch from -30 to 20, and Yaw from -200 to 200. Below these is a 3D visualization of the drone with a control interface. On the right, the 'Controller_UDP' Simulink model is shown, featuring a block diagram with a yellow 'UDP Receiver' block, a green 'UDP Transmitter' block, and a blue 'Parrot AR Drone' block. A summary and testing notes are visible in the bottom right of the Simulink window.

Summary:

- need to change a few parameters to accomplish hovering
- need to investigate offset value while drone propellers are running (does it change? is it cosine)
- need to parameterize more items (ie: nominal thrust)
- need to implement dynamic offset (possible dead zone?)
- need to see what data control gains have on stable altitude determination
- simulink projects, libraries, various subsystems. Try find better ways to implement subsystems/out

Testing needs:

- have math will get a soft cushion
- need to add repeatable / safe landing sequence

Altitude controller:

- had to change gains, different from last one
- nominal thrust was changed to be higher
- ???

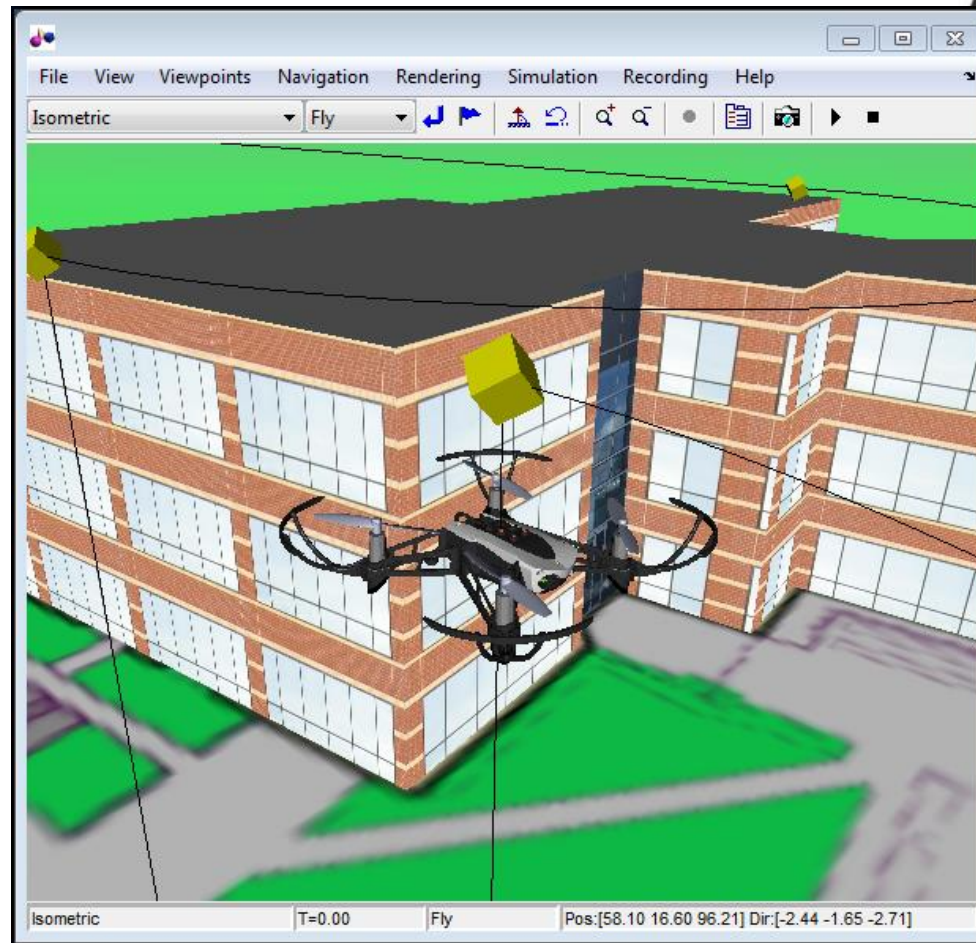
Parrot Mini Drone Support in Simulink



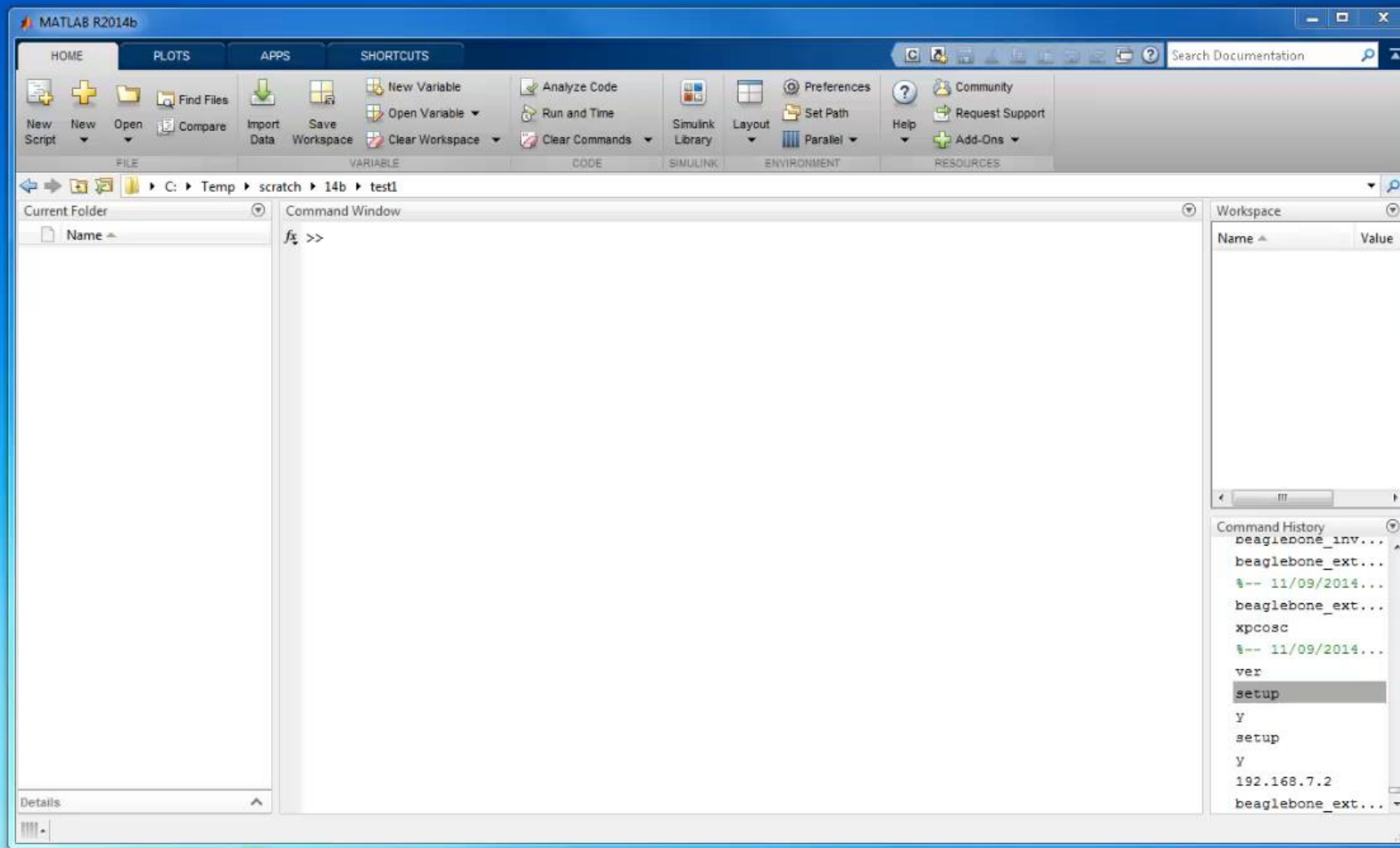
16.30 Feedback Control Systems

An MIT Feedback Control Systems Class that Teaches with Palm-size Drones

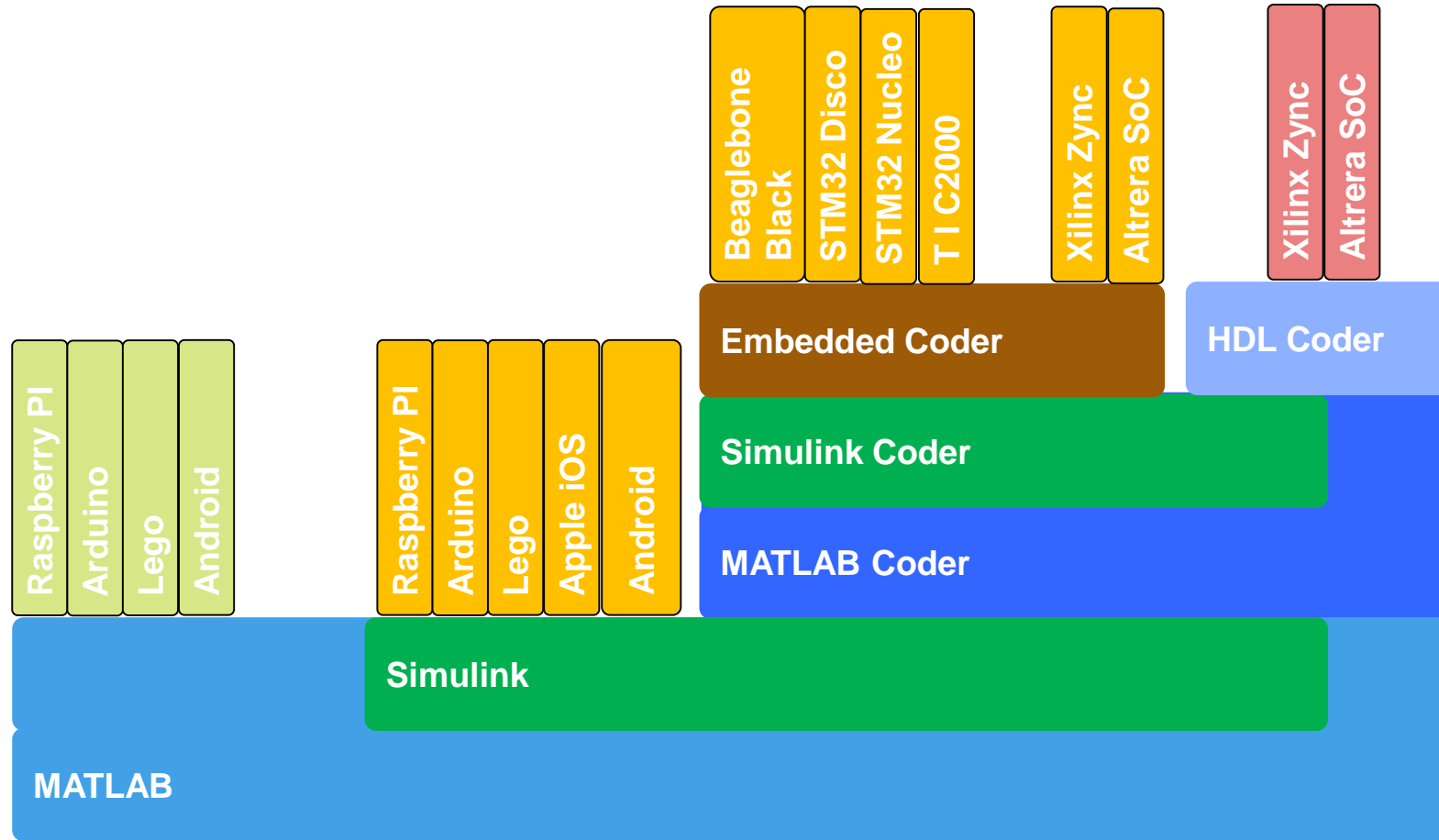
<http://fast.scripts.mit.edu/dronecontrol/>



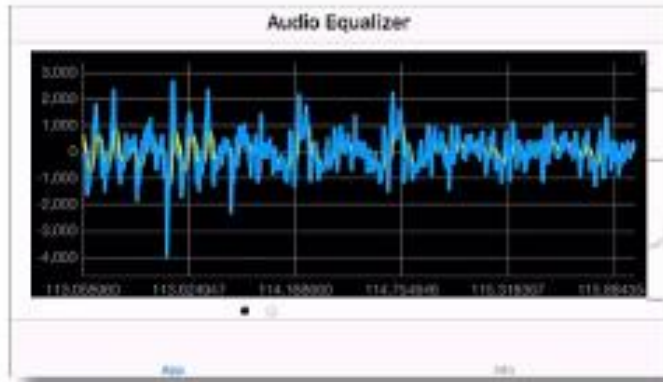
Using the Support Package Installer is Easy!



MathWorks Hardware Support Packages



iPhone iPad and Android Support



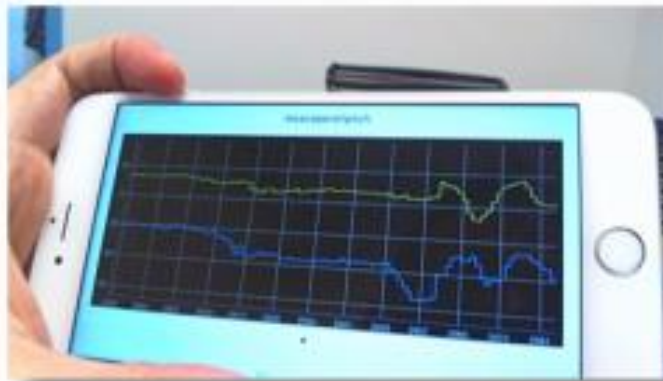
1) Audio processing



2) Image processing



3) Video processing



4) Mobile sensing



5) IoT with ThingSpeak



6) Wireless connectivity

Arduino board support from MATLAB and Simulink

2012b

UNO



MEGA



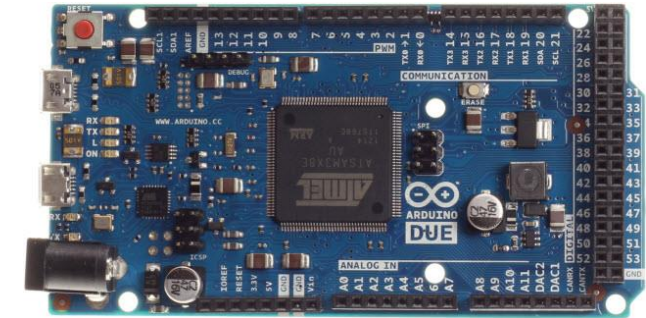
2013b

NANO



2014a

DUE

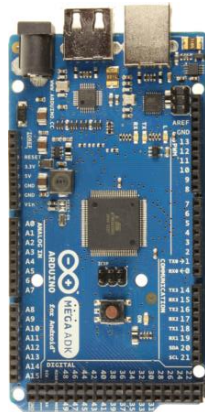


2014b

Leonardo



Mega ADK



Mini



Fio



Mac OS X Support!

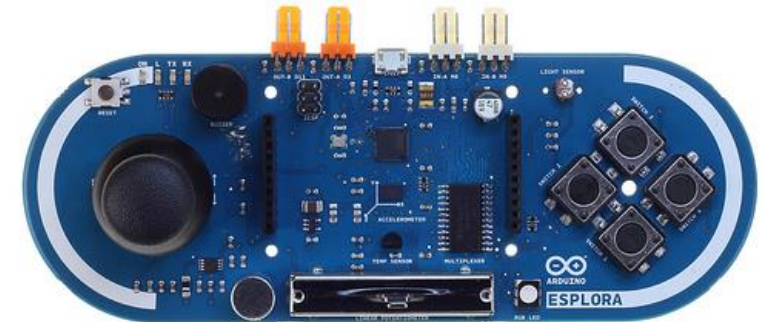
Pro



Micro



Esplora



Arduino Shields



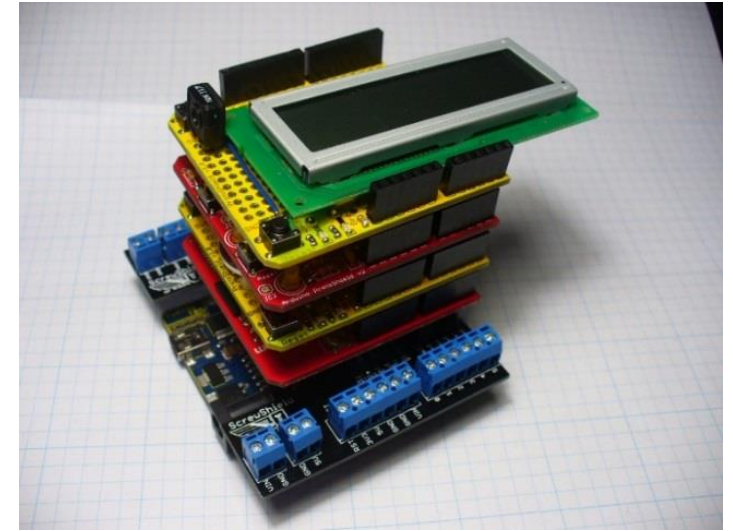
Ethernet Shield



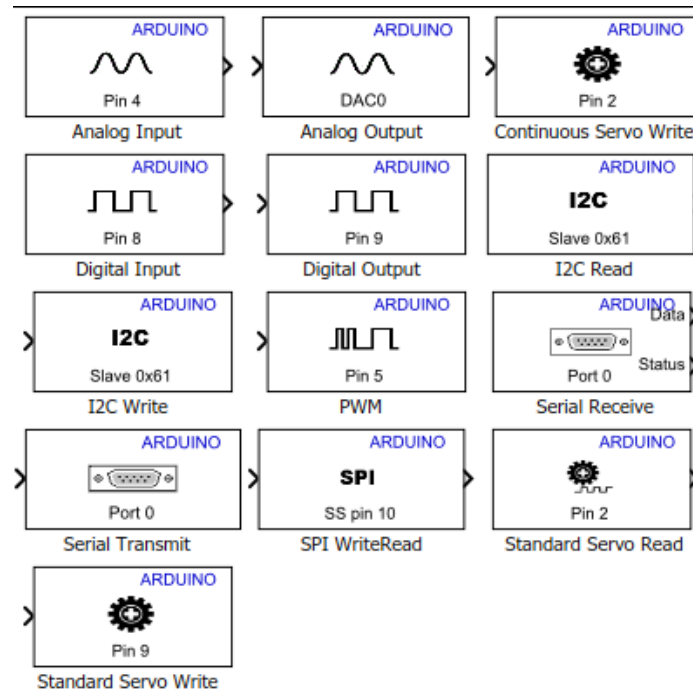
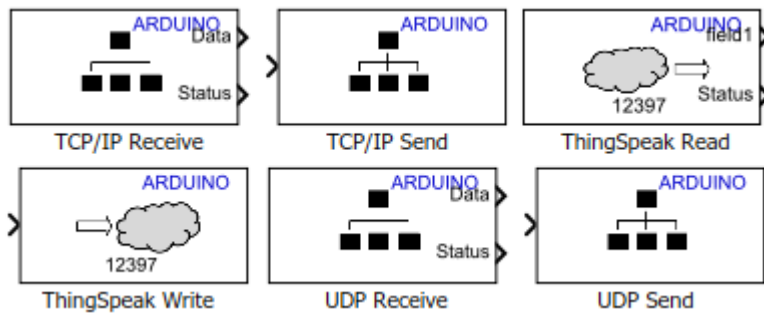
Wifi Shield



Motor Shield

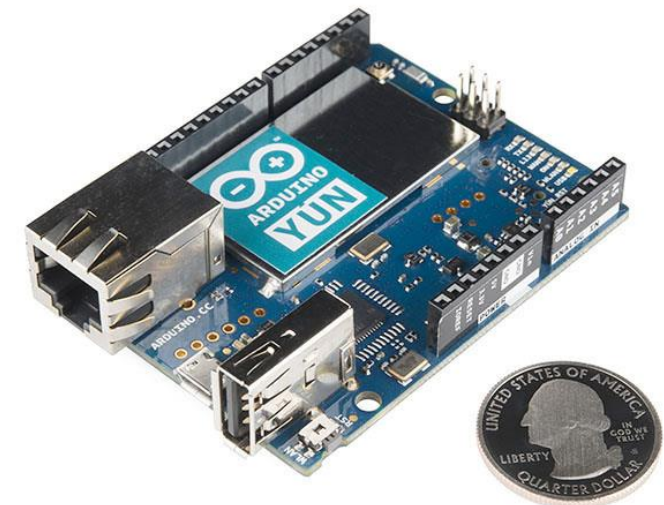


Simulink Blocks



Arduino

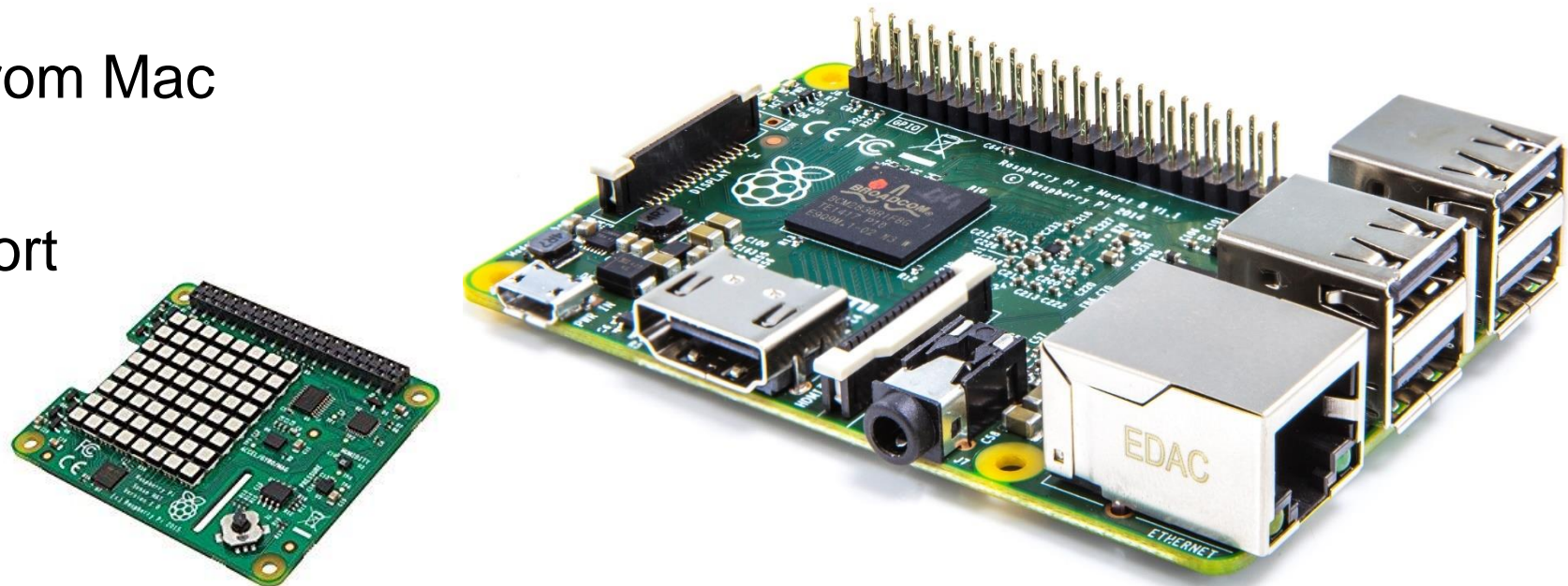
- **17a** – print/println support
- **16b** – Thingspeak read, PIL, enhanced external mode
- **15a** – Support from Linux
- **New IO Block support:**
 - I2C, SPI
 - UDP/TCP to LEGO, Raspberry Pi, and Android/iPhone
- **New boards: Yun**



Raspberry Pi

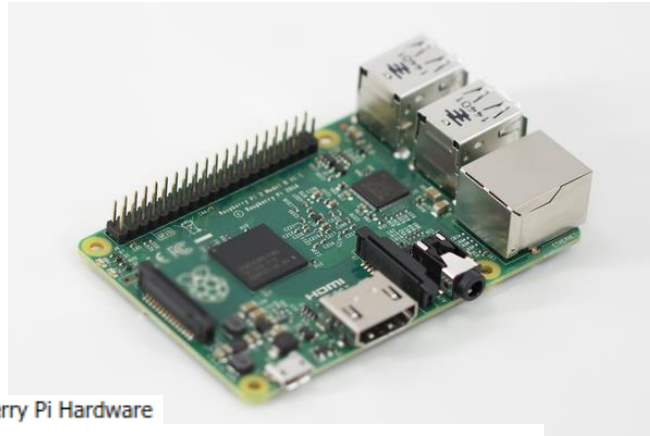
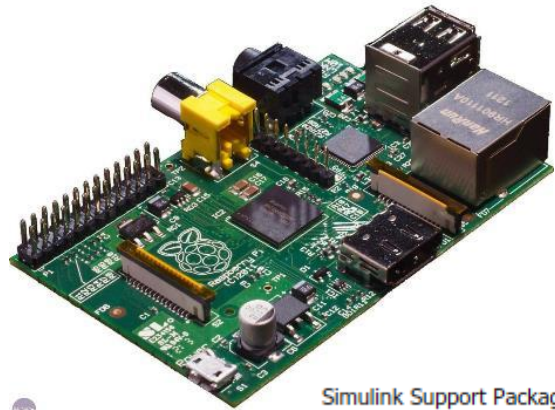
- 17a – New blocks: I2C, SPI, UART, TCP/IP, IMU, Pressure, Humidity, etc
- 16b - Support from Linux PC
- 16a – Pi 3 support
- 15b – Support from Mac
- 15a – Pi 2 support

**Thingspeak
Read/Write Support!**

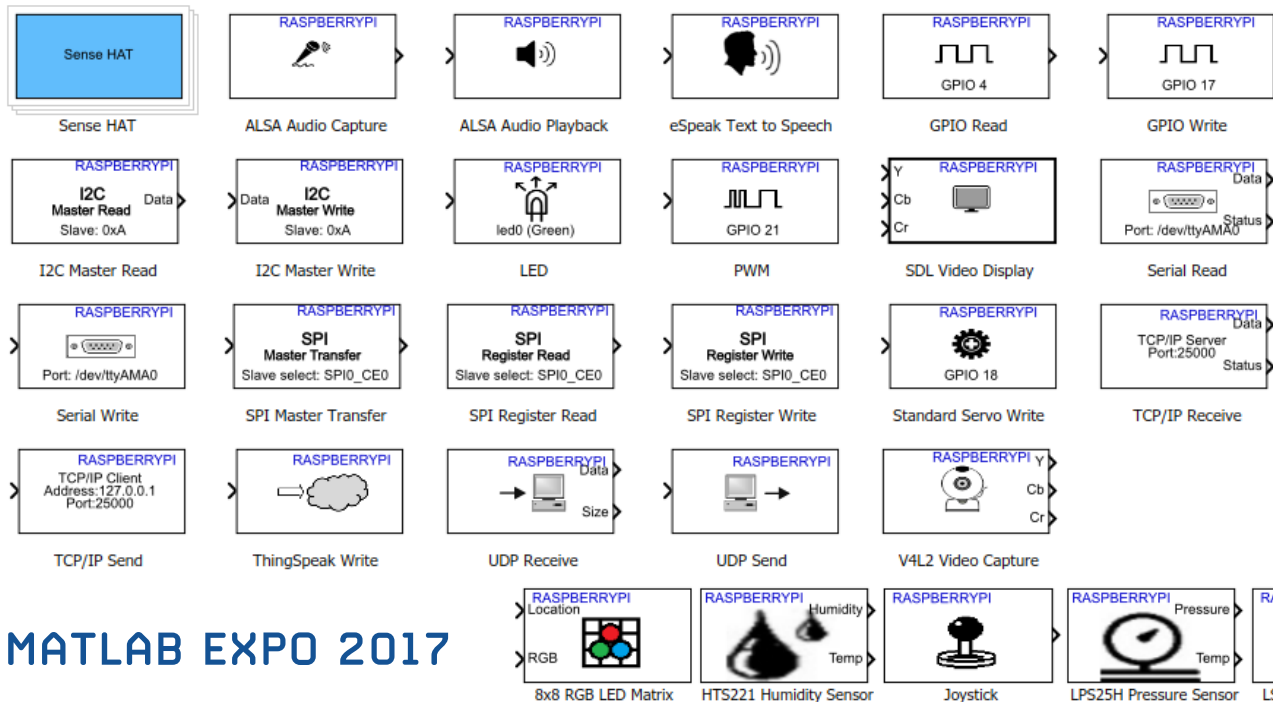


Linux Boards supported by MATLAB and Simulink

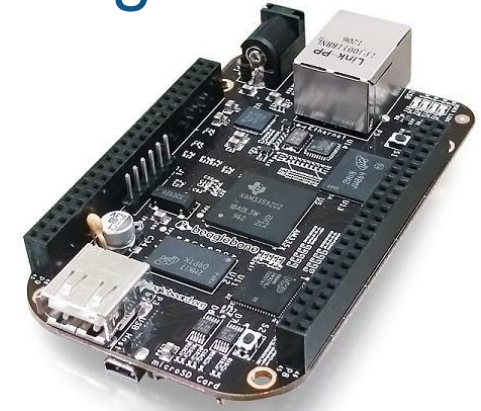
Raspberry Pi 1, 2, and 3



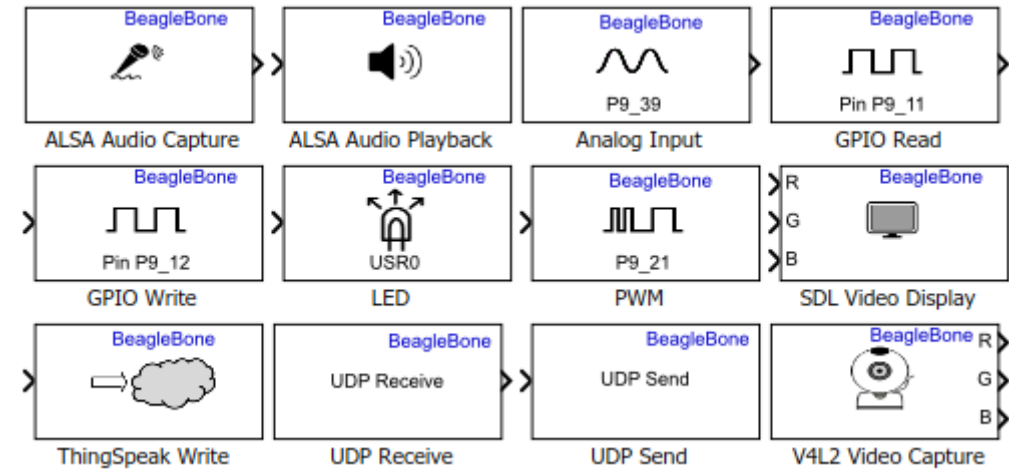
Simulink Support Package for Raspberry Pi Hardware



BeagleBone Black

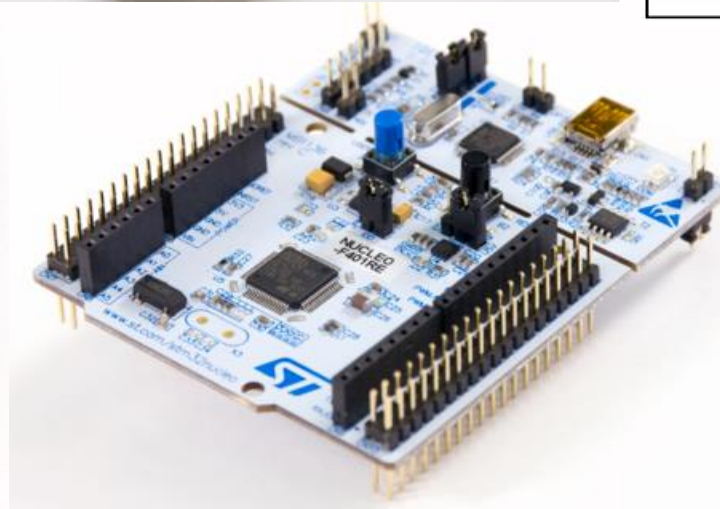
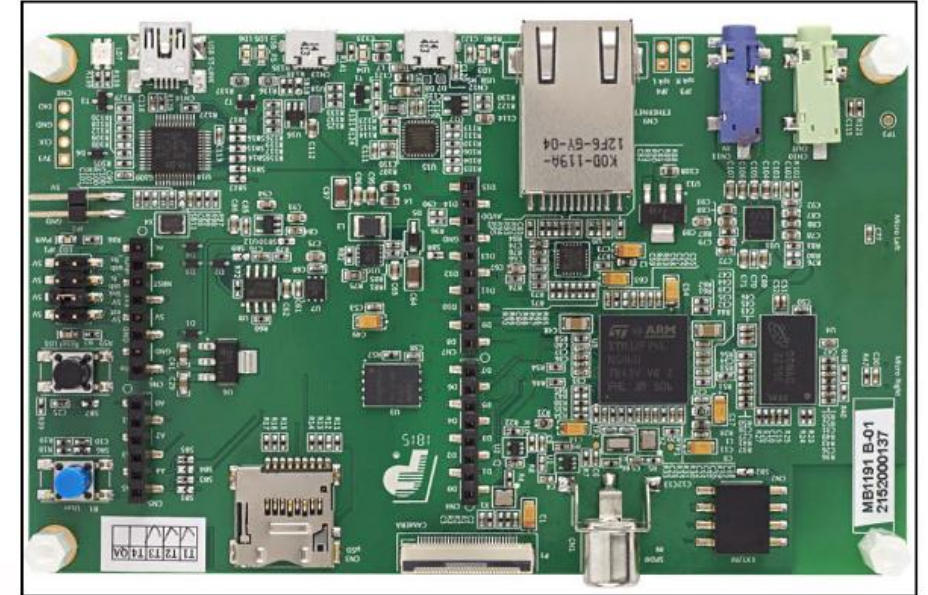


Embedded Coder Support Package for BeagleBone Black Hardware



STM32 Board Support

- STM32 Discovery : F407 & F746
- STM32 Nucleo :
 - STM32F746
 - STM32F411
 - STM32F401
 - STM32F302
 - STM32F103
 - STM32F031
 - STM32L476
 - STM32L053

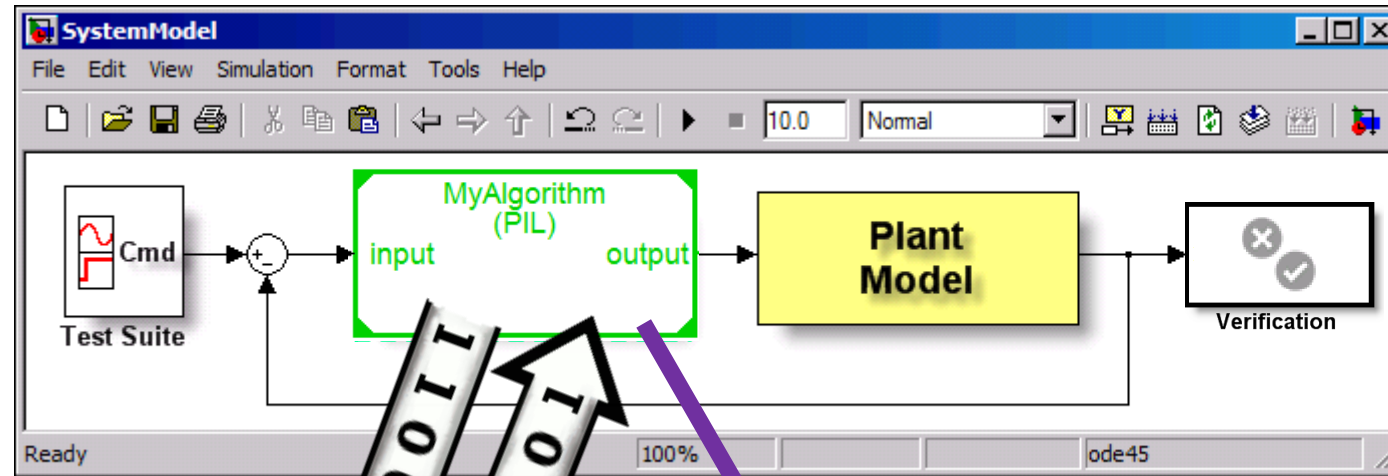


Agenda

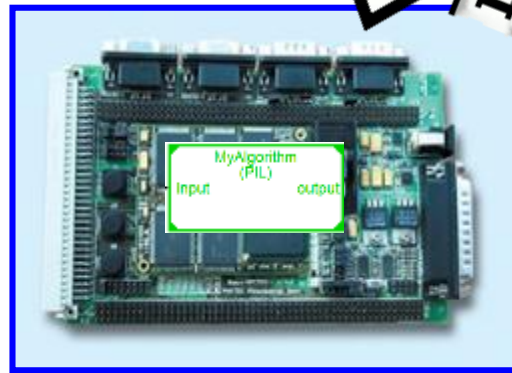
- Introduction
- Hardware Support Packages for MATLAB and Simulink
- **Processor-in-the-Loop Execution**
- Code Generation within the Internet of Things (IoT)
- Conclusion
- Questions

How SIL and PIL Work

On-Target Simulation

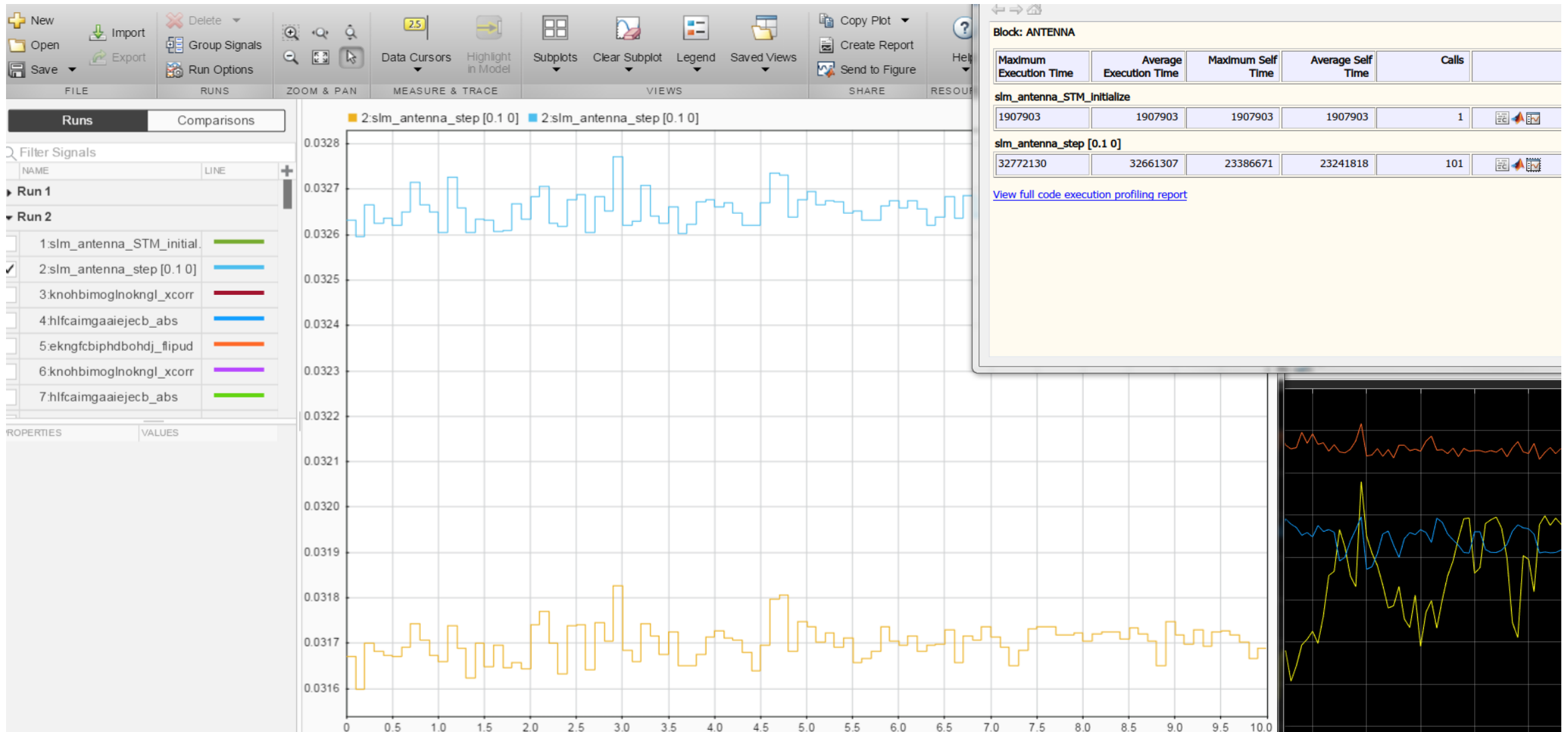


Non-Real-Time Synchronization with Host at Each Time Step

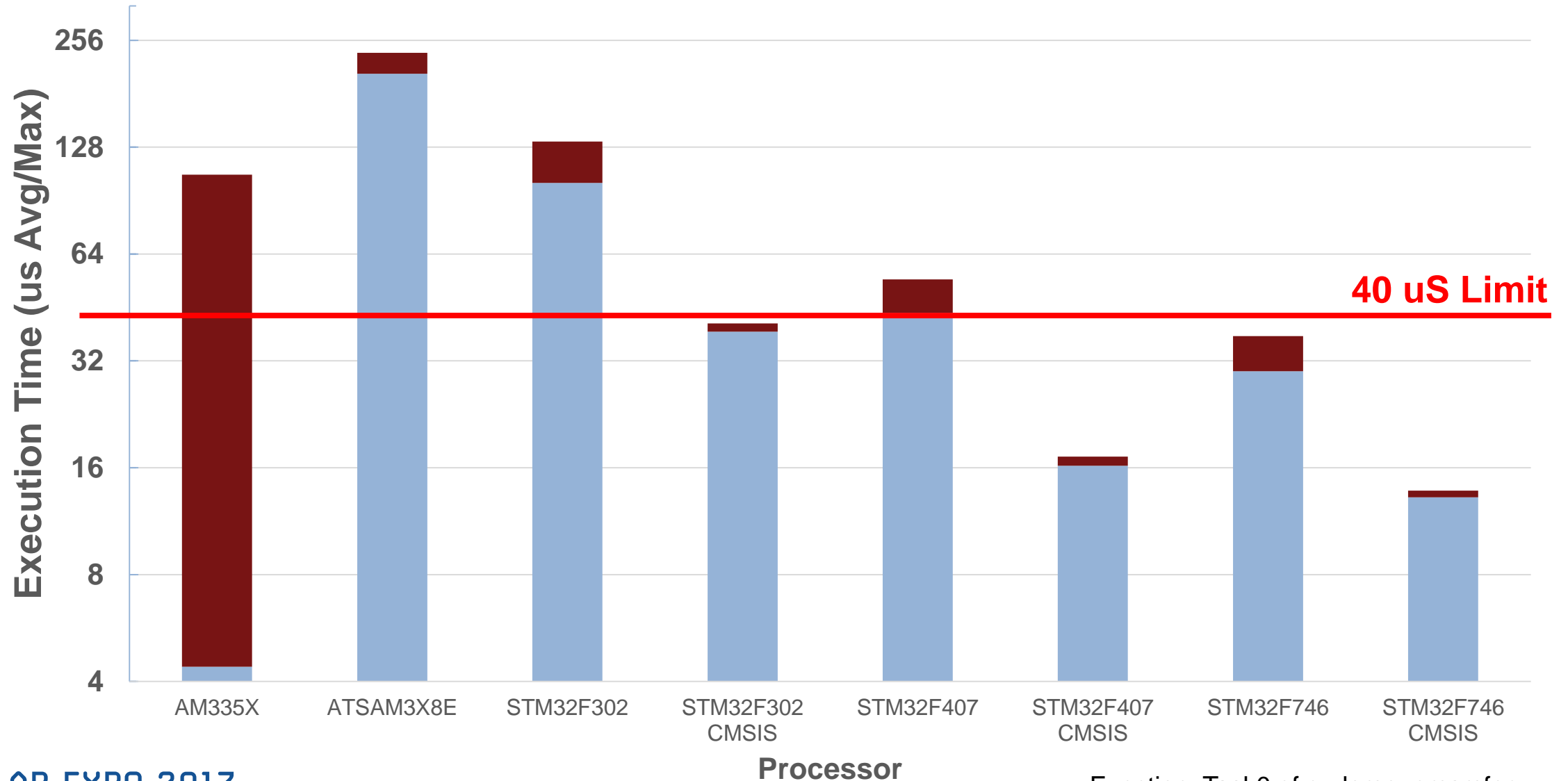


- Execution History**
- Equivalence comparison
 - Code coverage
 - Execution timing (profiling)

Processor-in-the-Loop (PIL) profiling

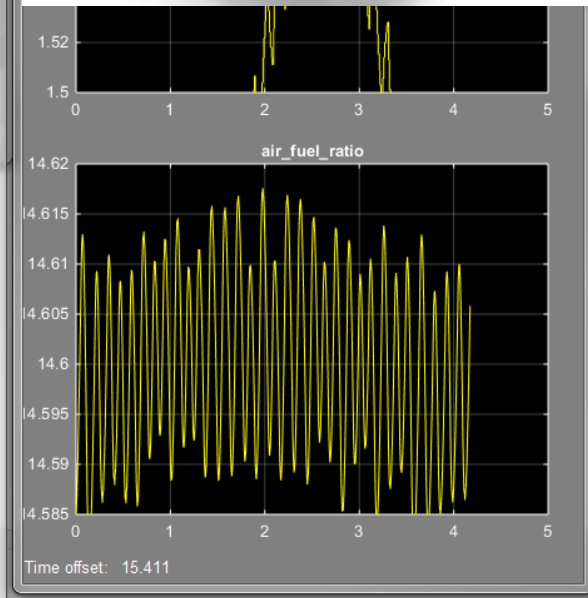


Processor Benchmarks on various ARM Cortex CPUs



Deploy Simulink Model to Beaglebone Black

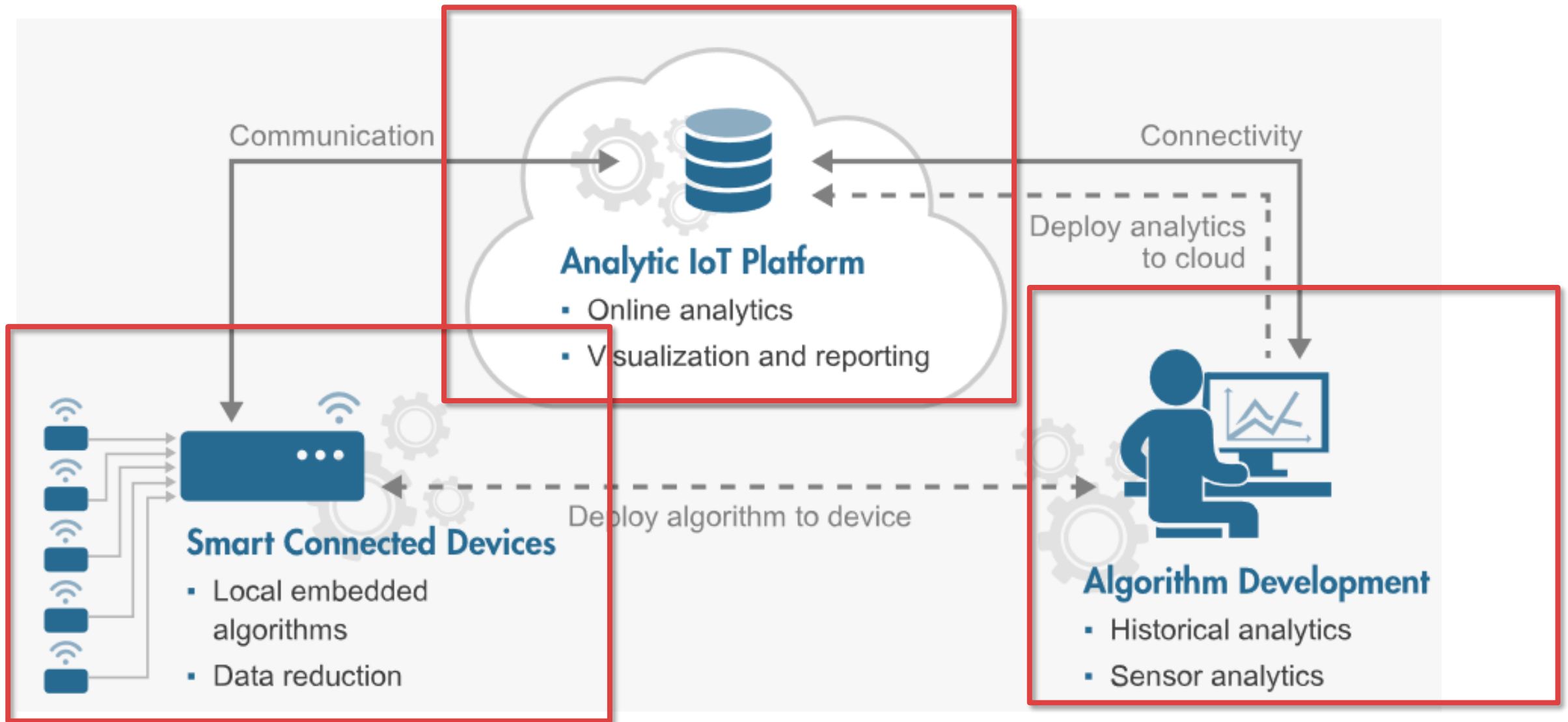
The image displays the Simulink environment with two main windows. The top window shows a state machine diagram for 'control_logic' with states: Fueling_Mode, Running, Low_Emissions (fuel_mode = LOW), Warmup, Rich_Mixture (fuel_mode = RICH), and Fuel_Disabled (fuel_mode = DISABLED). Transitions are labeled with 'DEC' and 'H'. The bottom window shows a block diagram titled 'Fault-Tolerant Fuel Control System'. It includes a 'Throttle Command' block, a 'Convert' block, 'sensors fuel_rate', another 'Convert' block, and an 'Engine Gas Dynamics' block. The system also features a 'Throttle Sensor', 'Speed Sensor', 'ego', 'EGO Sensor', and 'MAP Sensor'. A note at the bottom states: 'The sensor switches simulate any combination of sensor failures. The Engine Speed Selector switch simulates different engine speeds (rad/sec). Copyright 1990-2014 The MathWorks, Inc.'



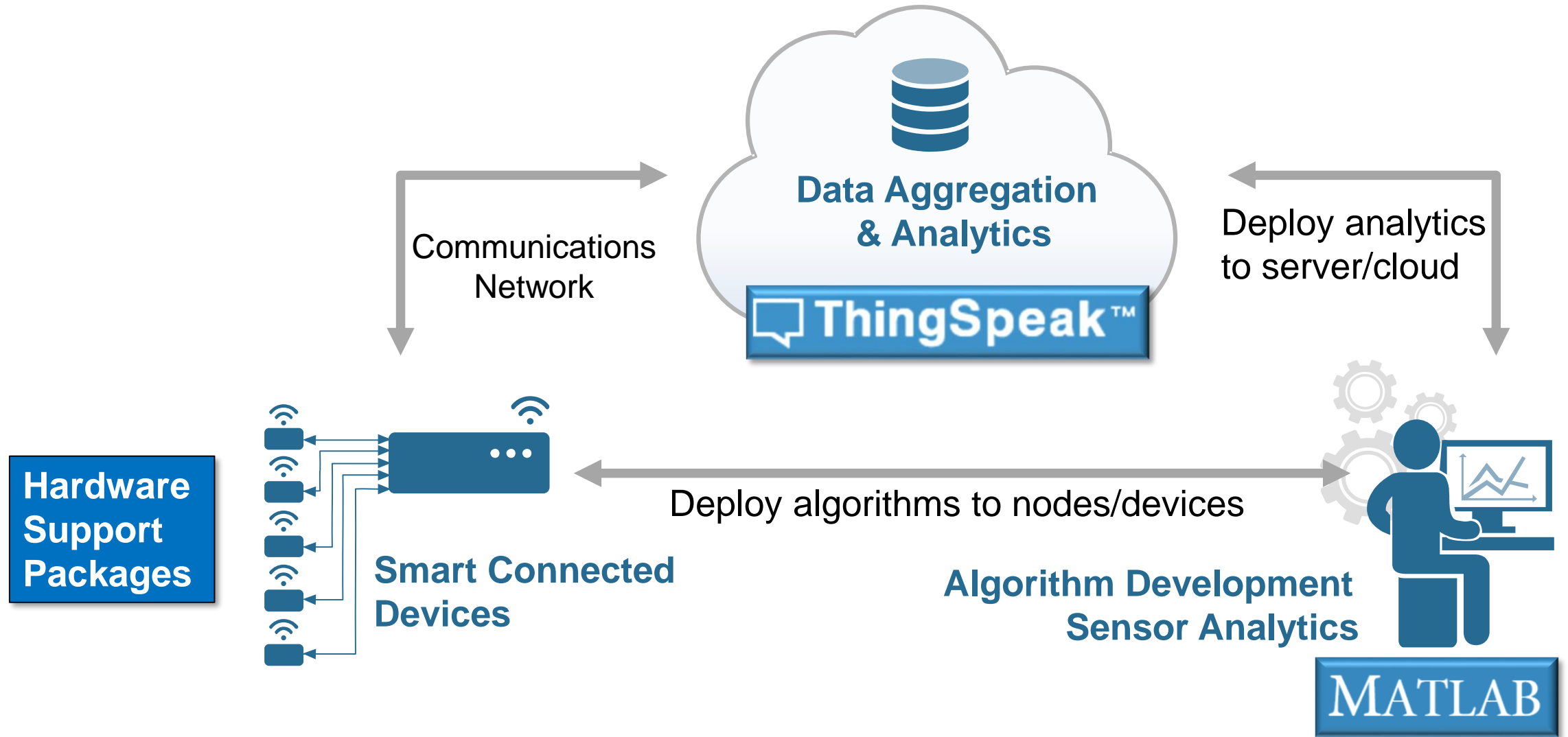
Agenda

- Introduction
- Hardware Support Packages for MATLAB and Simulink
- Processor-in-the-Loop Execution
- **Code Generation within the Internet of Things (IoT)**
- Conclusion
- Questions

IoT Analytics Challenges

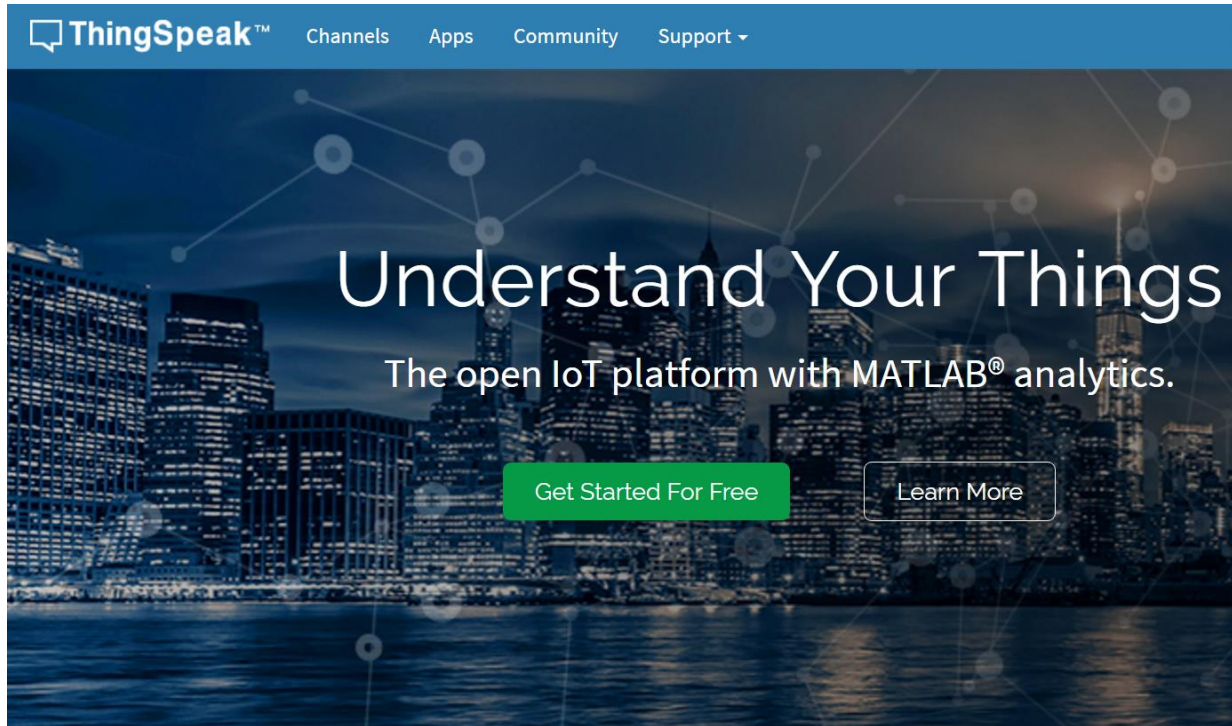


IoT Analytics Solutions



What Is ThingSpeak?

Web Site For People



Web Service for Devices

```

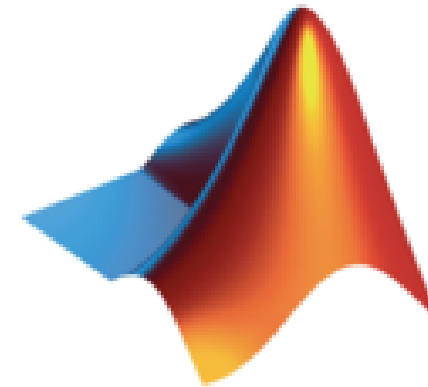
{
  - channel: {
    id: 38629,
    name: "Car Counter",
    description: "Counting number of cars passing a reference line in 15 sec interval",
    latitude: "42.28",
    longitude: "-71.35",
    field1: "Number of Westbound Cars",
    field2: "Number of Eastbound Cars",
    created_at: "2015-05-19T20:14:03Z",
    updated_at: "2016-05-19T10:36:35Z",
    last_entry_id: 1477231
  },
  - feeds: [
    - {
      created_at: "2016-05-19T10:36:20Z",
      entry_id: 1477230,
      field1: "18.000000",
      field2: "8.000000"
    },
    - {
      created_at: "2016-05-19T10:36:35Z",
      entry_id: 1477231,
      field1: "18.000000",
      field2: "14.000000"
    }
  ]
}

```

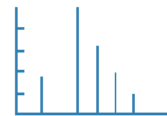
ThingSpeak

- New MathWorks web service hosted on AWS
- Lets you collect, analyze and act on data from “things”
- Over **130,000** users worldwide
- It has **MATLAB** for IoT Analytics
- It's **free** to get started

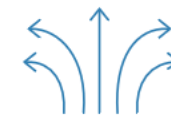
<https://thingspeak.com>



Collect

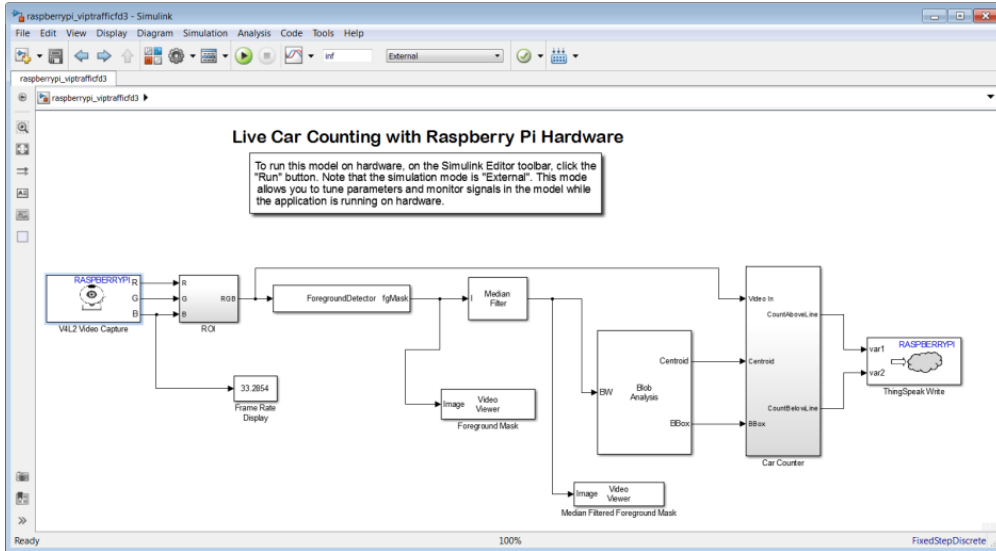


Analyze



Act

Car-counting camera IoT example



ThingSpeak Channels ▾ Apps Blog Support ▾

Apps / MATLAB Analysis / Calculate daily volume for previous 24 hours

Name

MATLAB Code

```

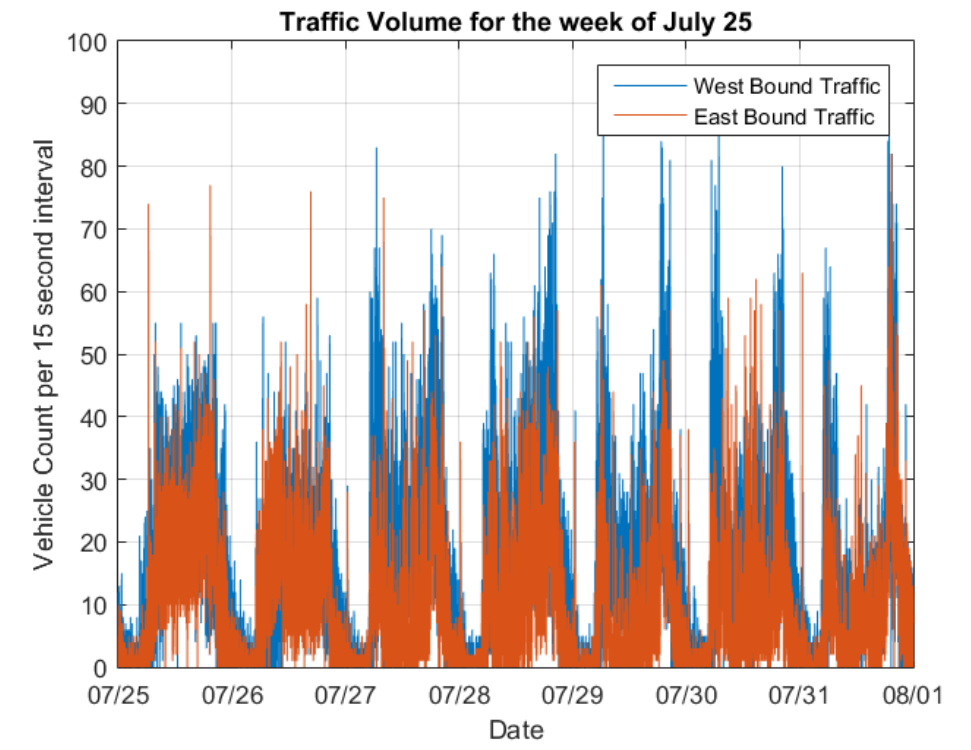
1 % Enter your MATLAB Code below
2 startTime = datetime((now-1), 'ConvertFrom', 'datenum');
3 stopTime = datetime(now, 'ConvertFrom', 'datenum');
4 startDate = datetime(startTime, 'InputFormat', 'MMMM d, yyyy HH:mm:ss ');
5 endDate = datetime(stopTime, 'InputFormat', 'MMMM d, yyyy HH:mm:ss ');
6 datevector = [startDate, endDate];
7 [DailyEast, t] = thingSpeakRead(38629, 'Fields', [2], 'DateRange', datevector, 'URL', 'https://api.thingspeak.com/');
8 DailyVolume = sum(DailyEast)
9 thingSpeakWrite(51671, DailyVolume, 'URL', 'https://api.thingspeak.com/', 'WriteKey', 'XXXXXXXXXX')
10
                    
```

Run and Save

Output

DailyVolume =

43386



Custom Visualizations with ThingSpeak- Weather Station Example

Natick Weather: Median Temp Overlay

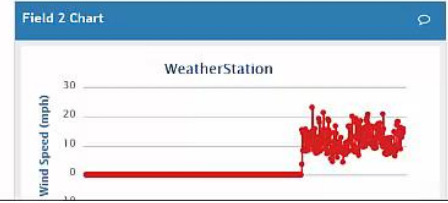
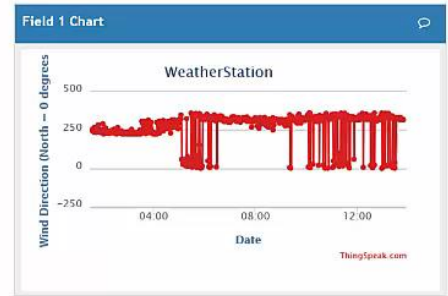
Natick Median Temperature, past 3 days

Time	Temperature (°F)
mid	37
12am	38
3am	39
6am	40
9am	42
12pm	43
3pm	42
6pm	40
9pm	38
mid	37

Channel ID: 12397
 Author: hemdanw
 Access: Public

MathWorks Weather Station, West Garage, Natick, MA 01760, USA
 MathWorks Weather Station, weather, MathWorks

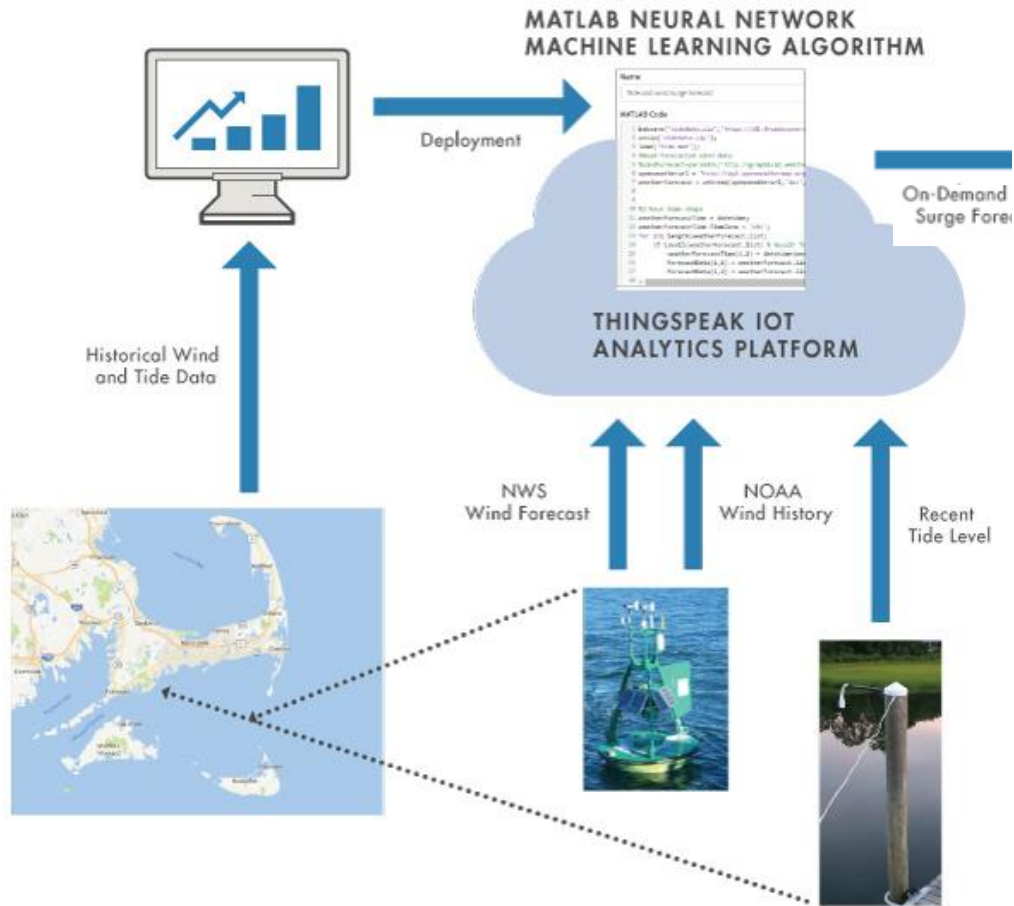
Buttons: Data Export, MATLAB Analysis, MATLAB Visualization



Channel Status Updates

Placeholder for status updates.

Predictive Analytics Example with ThingSpeak



ThingSpeak™ Channels Apps Community Support How to Buy Account

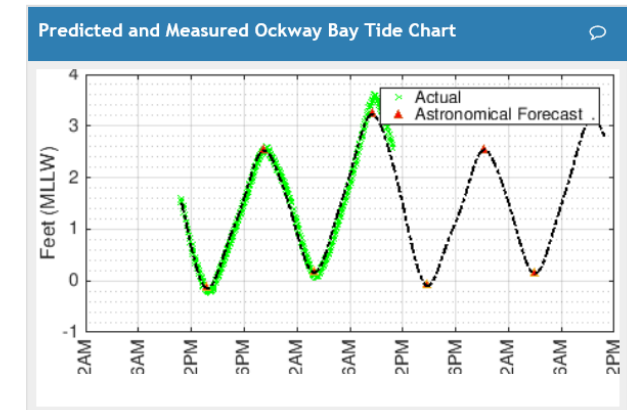
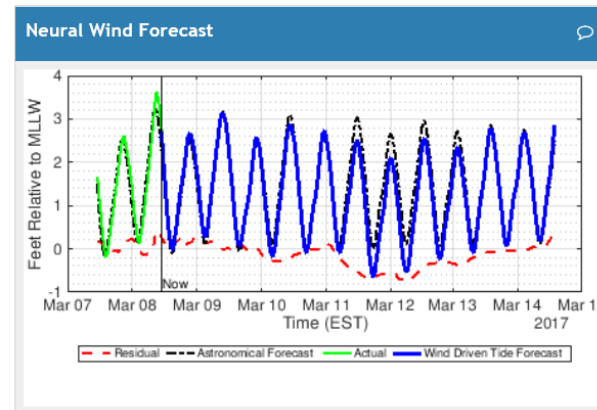
Predicted and Measured Ockway Bay Tide Chart

Channel ID: 137305
 Author: mawrey
 Access: Public

Tide measurement and forecasting with the effect of wind predicted using neural networks.
 #tide, #wind surge, #neural network

Data Export More Information

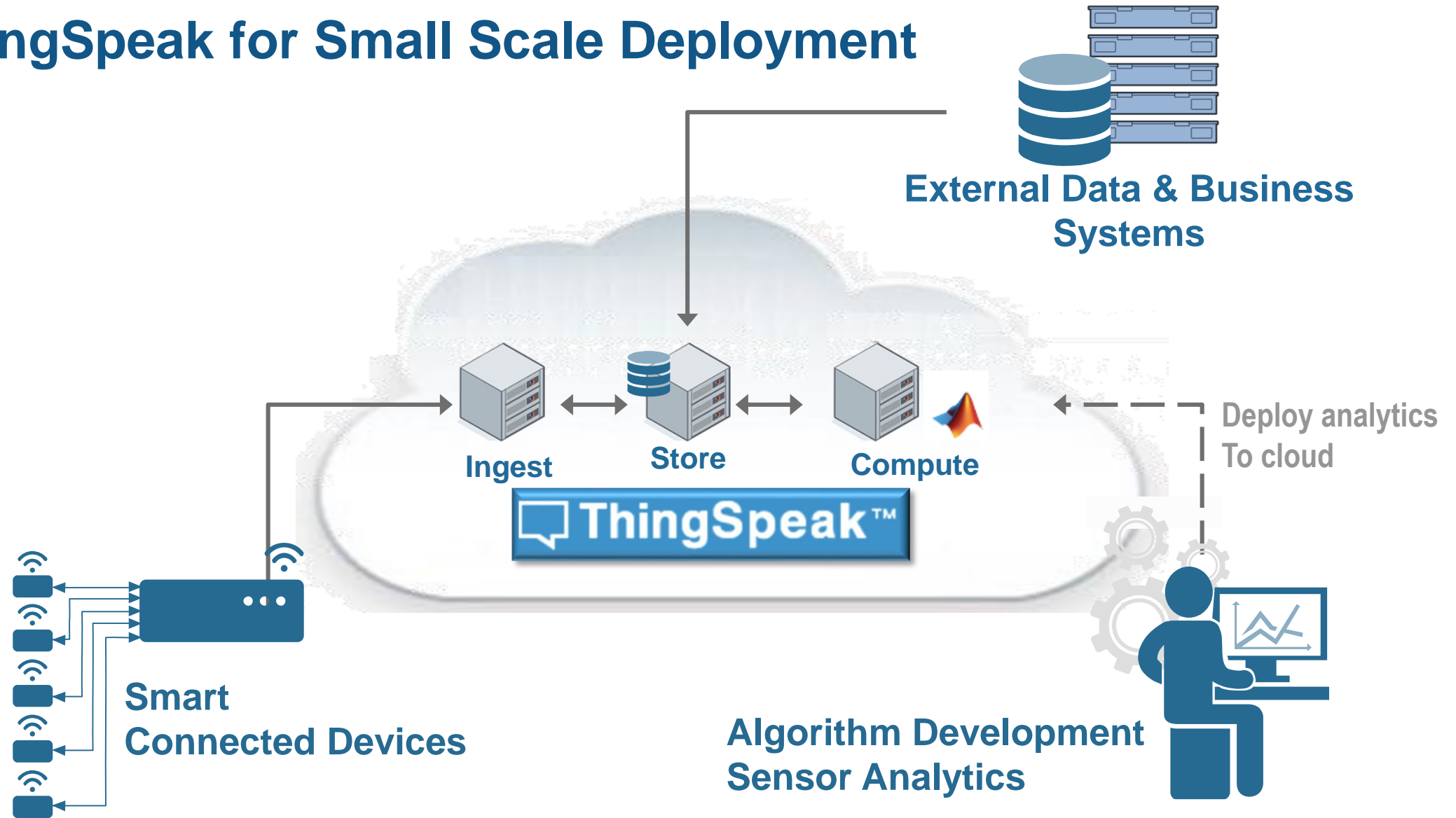
MATLAB Analysis MATLAB Visualiz



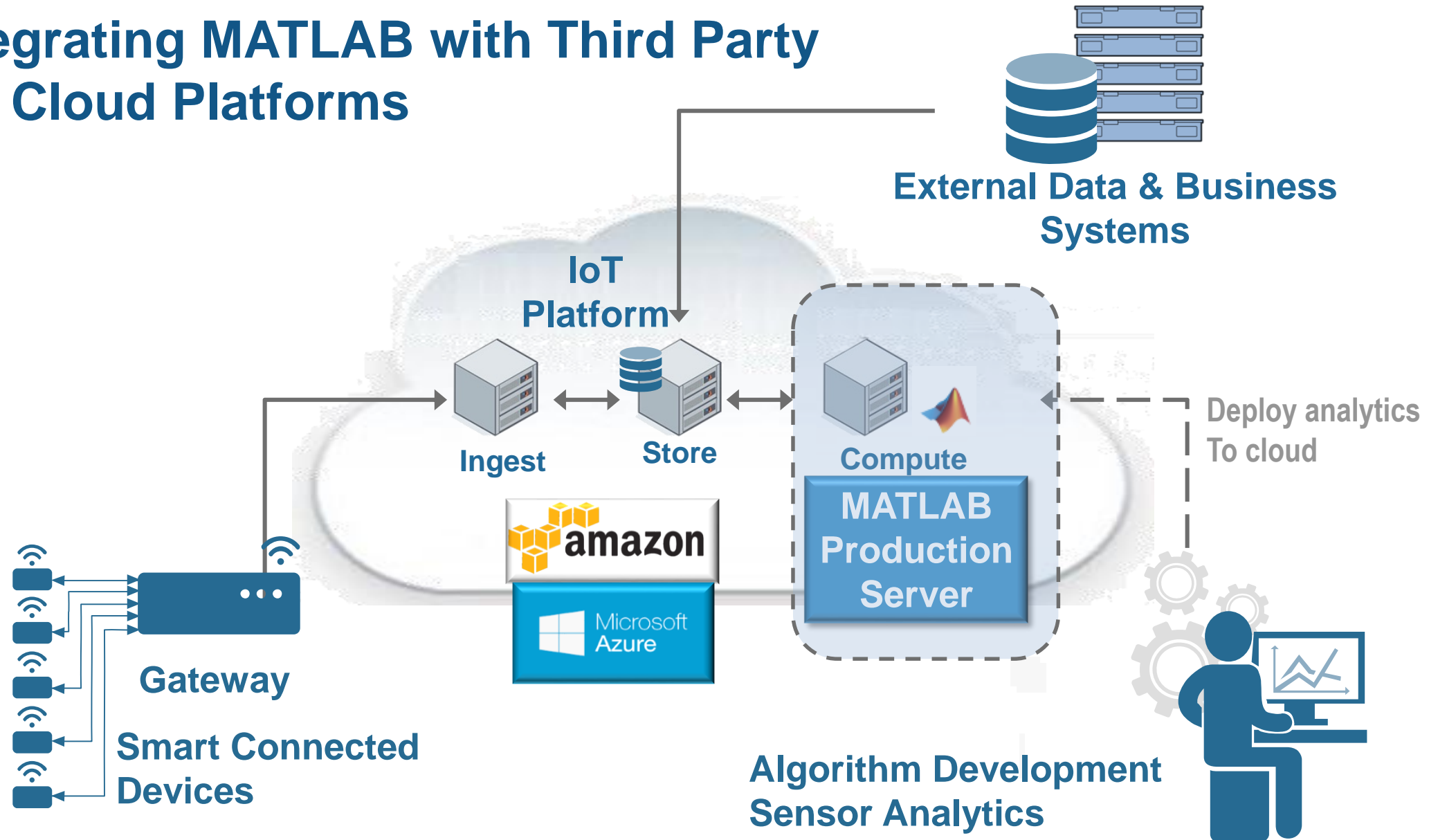
Residual Astronomical Forecast Actual Wind Driven Tide Forecast

```
netWind.divideFcn = 'dividerand'; % Divide data randomly
netWind.divideMode = 'sample'; % Divide up every sample
netWind.divideParam.trainRatio = 70/100;
netWind.divideParam.valRatio = 15/100;
netWind.divideParam.testRatio = 15/100;
```

ThingSpeak for Small Scale Deployment



Integrating MATLAB with Third Party IoT Cloud Platforms

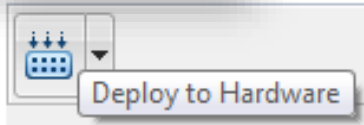


Agenda

- Introduction
- Hardware Support Packages for MATLAB and Simulink
- Processor-in-the-Loop Execution
- Code Generation within the Internet of Things (IoT)
- **Conclusion**
- Questions

Key takeaways

Hardware support in MATLAB and Simulink ...



- Code generation for prototype or production workflows
- Hardware Support Packages make it **easy to install and configure** the necessary software
- Supports many **Open Hardware Revolution** boards and mobile devices (iOS, Android)
- Enables smart sensors for the Internet of Things



Q&A

Déploiement embarqué et connectivité hardware avec MATLAB et Simulink

