Embedded Systems

Fatima Anwar
Networked & Embedded Systems Lab
Aug 2-3, 2017

Note: modified from original LACC 2016 slides by Paul Martin

Variability Expedition

UCLA

We typically think of computers as...

www.amazon.com

Embedded systems that assist

- RADAR & ultrasonic sensors
- Compute distances & localize
- Assist users who are tired

1.http://www.daimler.com/dccom/0-5-1210218-1-1210321-1-0-0-1210228-0-0-135-0-0-0-0-0-0-0-0.html
2. http://www.planetbenz.com/category/news/page/200/

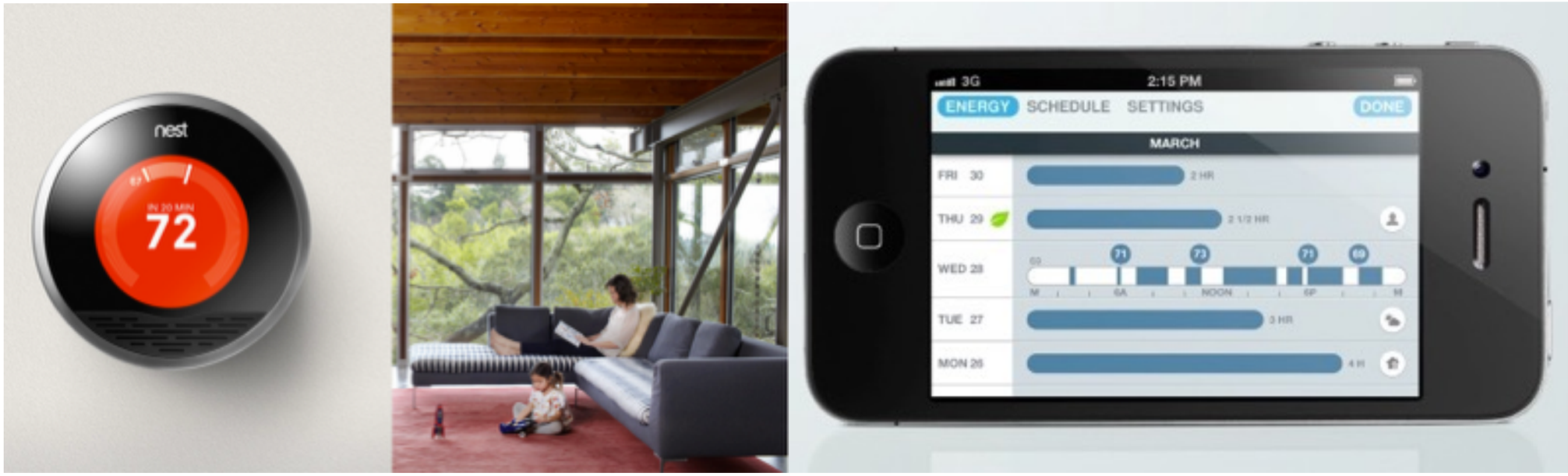
Mercedes: Learns how you drive

http://www.youtube.com/watch?v=A66zgJ4Oj8o&list=UUfRUa1Z5gTknsMMaKLthZIg&index=21&feature=plpp_video

Questions

- What are some of the key components that make this possible?
- What are some of the critical operations of this system?

Embedded systems that monitor



- Measure environmental factors
- Quietly adapt & control
- Provide real time data to the users

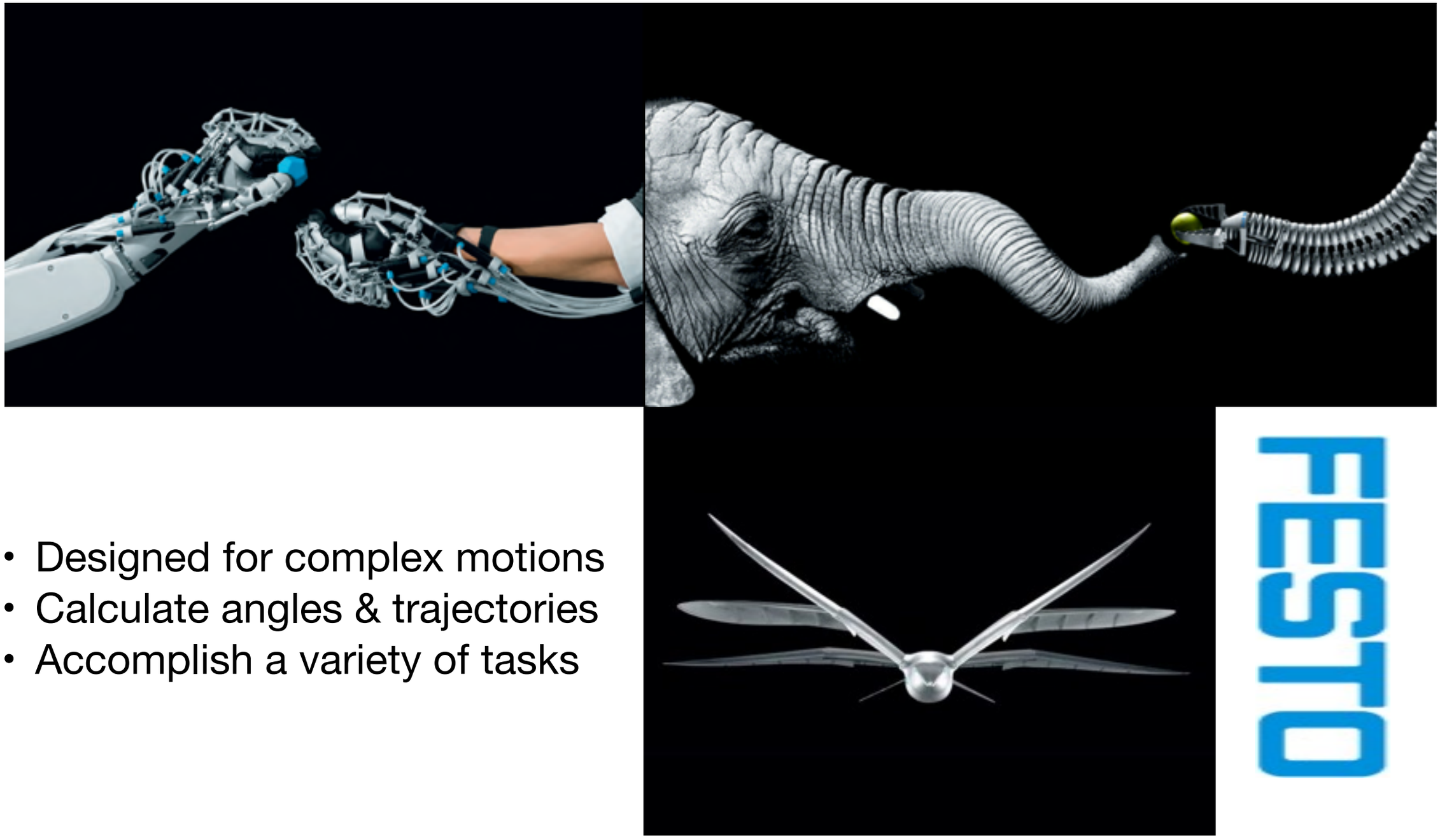
NEST: The Learning Thermostat

https://youtu.be/L8TkhHgkBsg

Questions

- How is this user interaction different from the Mercedes?
- What kinds of components might you need to make this work?

Embedded systems that control



- Designed for complex motions
- Calculate angles & trajectories
- Accomplish a variety of tasks

FESTO: Robots that mimic

Vote on a video:

Bird

Kangaroo

Aqua Ray

Air Ray

Jellyfish

Are you sure? The other ones are cooler…

Questions

- Why do we mimic nature?
- When does it make sense to mimic nature? When does it not?
- What are some common devices or inventions that mimic nature?

Geckos help clean space junk

https://www.facebook.com/pg/ajplusenglish/videos/?ref=page_internal

Embedded systems that care

- Food insecurity, supply shortage, difficulty in communication
- Use embedded systems to better distribute resources

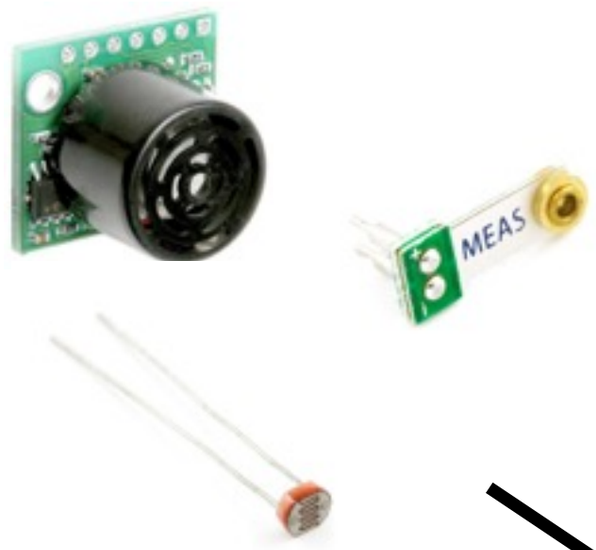
Embedded System or Cyber-Physical Systems

A Definition:

"A system of collaborating computational elements that sense or control physical entities"

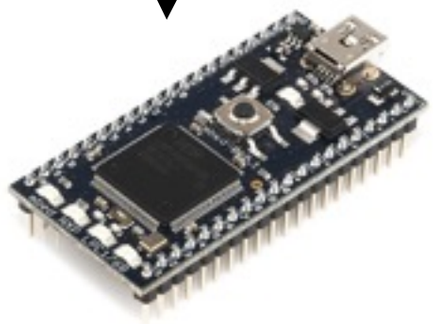
Embedded Systems Can do Anything!

Sensors



Communication

Actuators



1.All images from www.Sparkfun.com

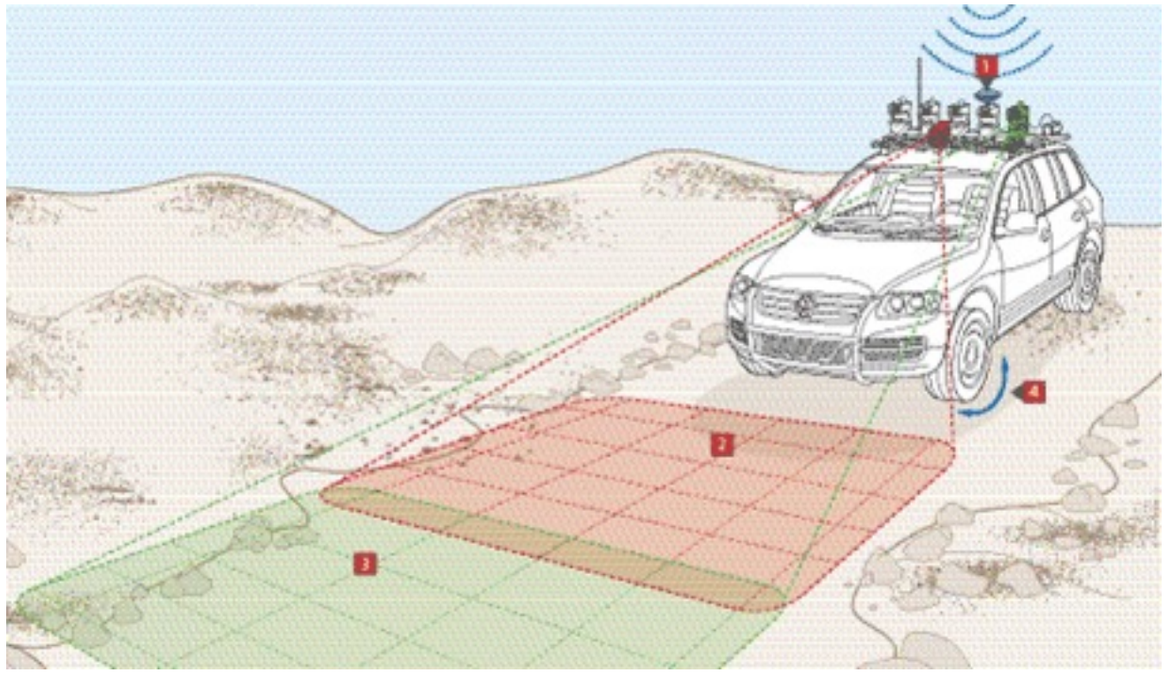
Computer / Processor

Stanford's Stanley: Autonomous Vehicle

- GPS
- SICK Laser Scanner
- Stereo Camera
- Monocular Camera
- 6DOF Inertial Sensors
- RADAR
- Speed sensor
- Pentium M
- Drive-by-Wire
- Battery System

1. http://cs.stanford.edu/group/roadrunner//old/technology.html

Stanford's Stanley: Probabilistic Terrain Analysis



(a) Beer Bottle Pass (b) Map and GPS corridor

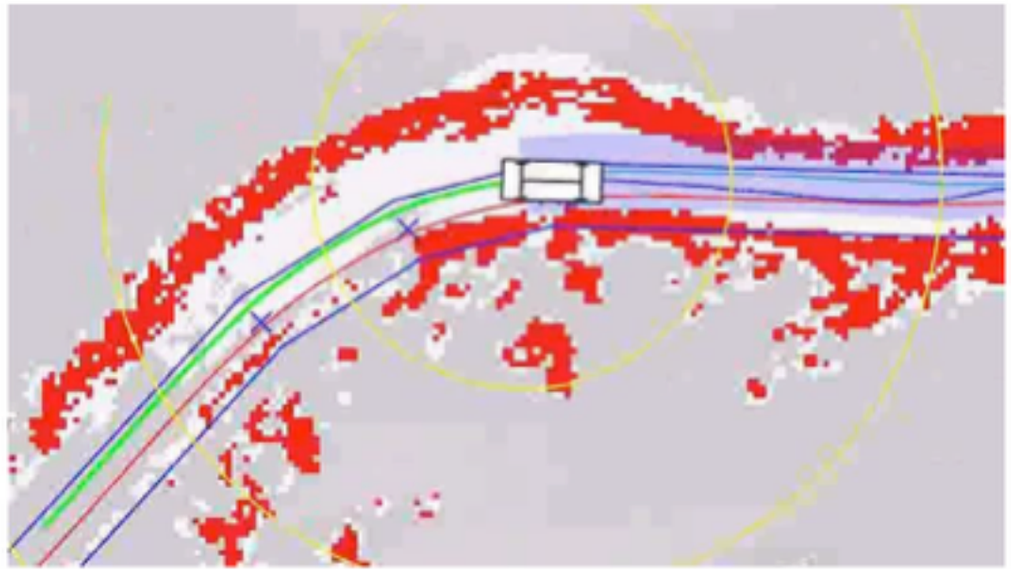
Scan terrain and collect data

↓

Run through probabilistic models

↓

Calculate paths & obstacles



red: obstacles
clear: paths
gray: unknown

Tracking Cranes

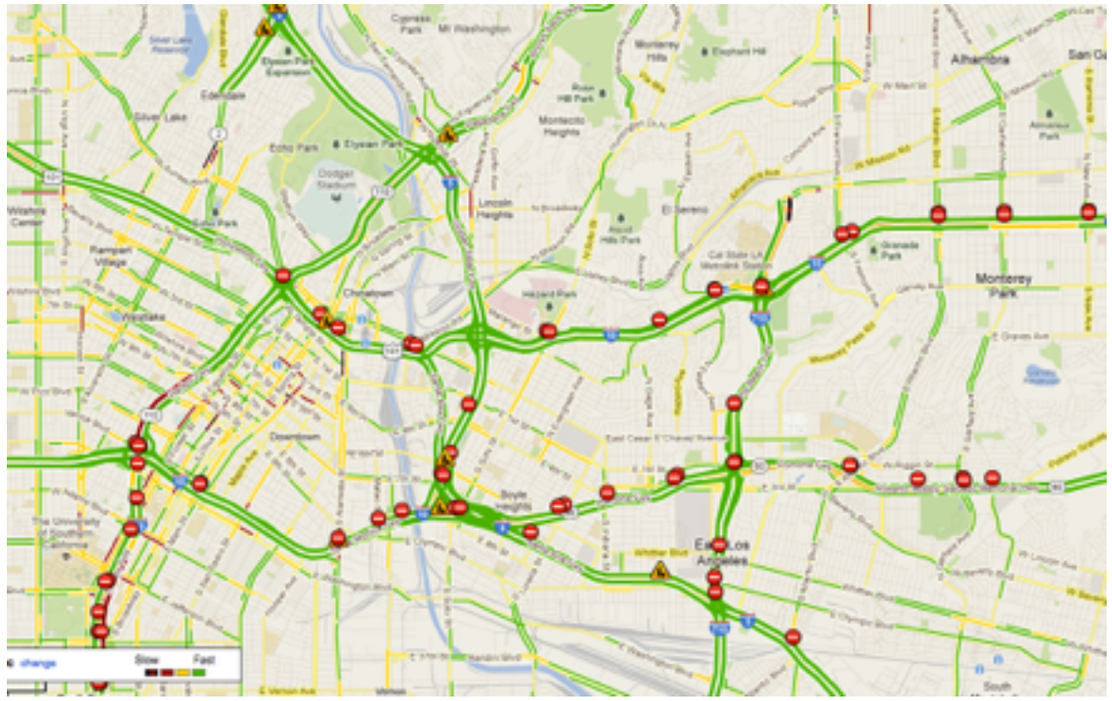
?

?

cellular + shortwave radio

University of Nebraska: Cyber Physical Networking Lab

Santa Monica Parking Sensors

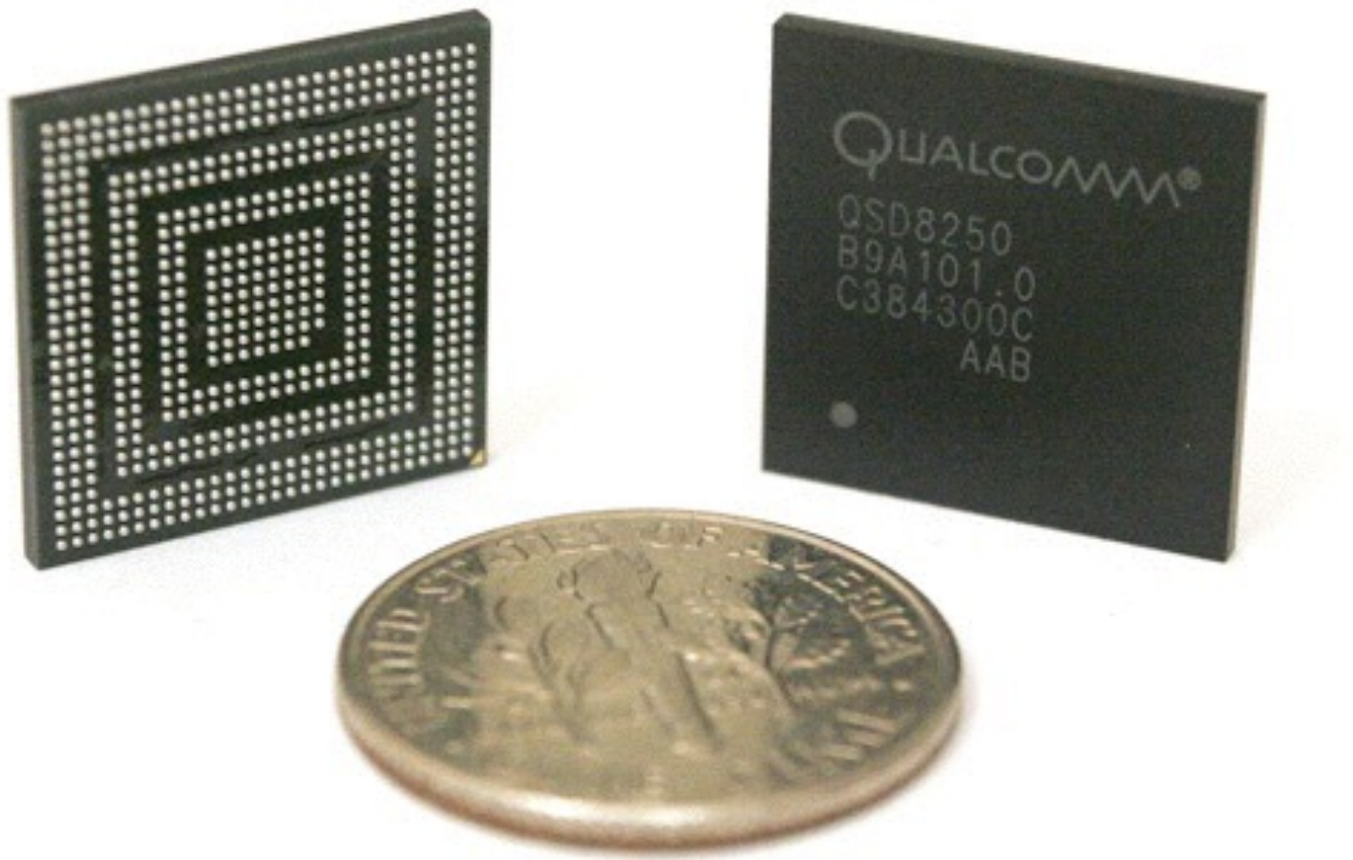


Information at your fingertips...

parking in motion
a smarter way to park

Embedded Computing

Embedded Processor



Embedded Processor: Comparison

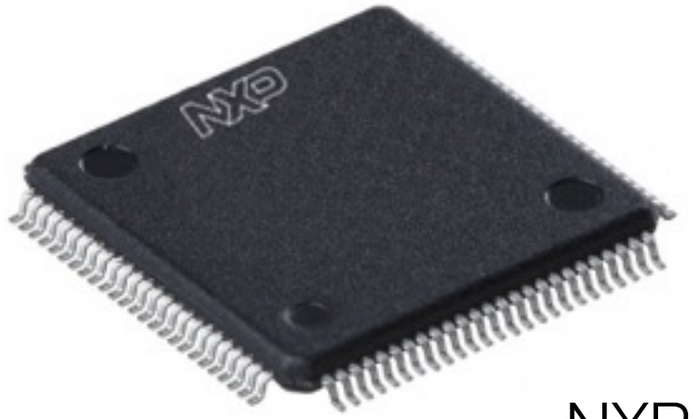
Intel Core i7

- Speeds: 3-4 GHz
- Cores: 4+
- Memory DDR3: 32 GB
- Power: 50 ~300 Watts

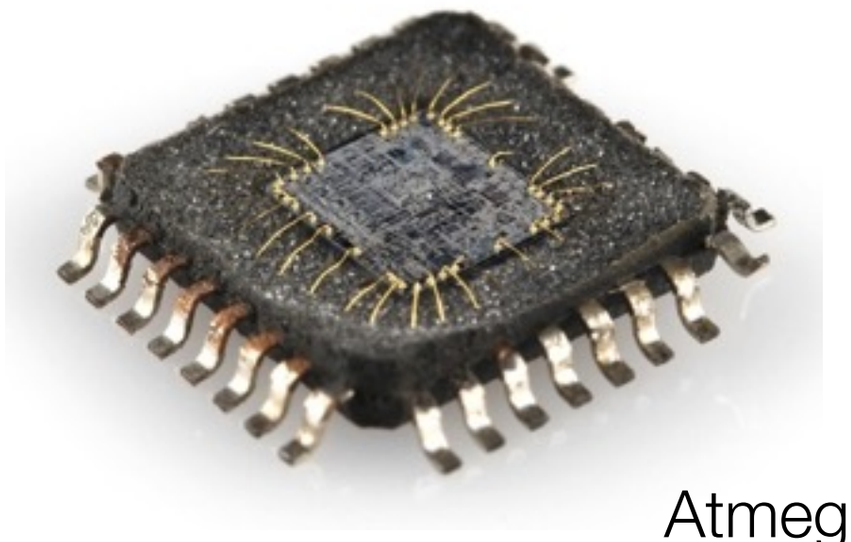
ARM Cortex-M3

- Speeds: 100 MHz
- Cores: 1
- Memory SRAM: 64 KB
- Power: ~0.5 W

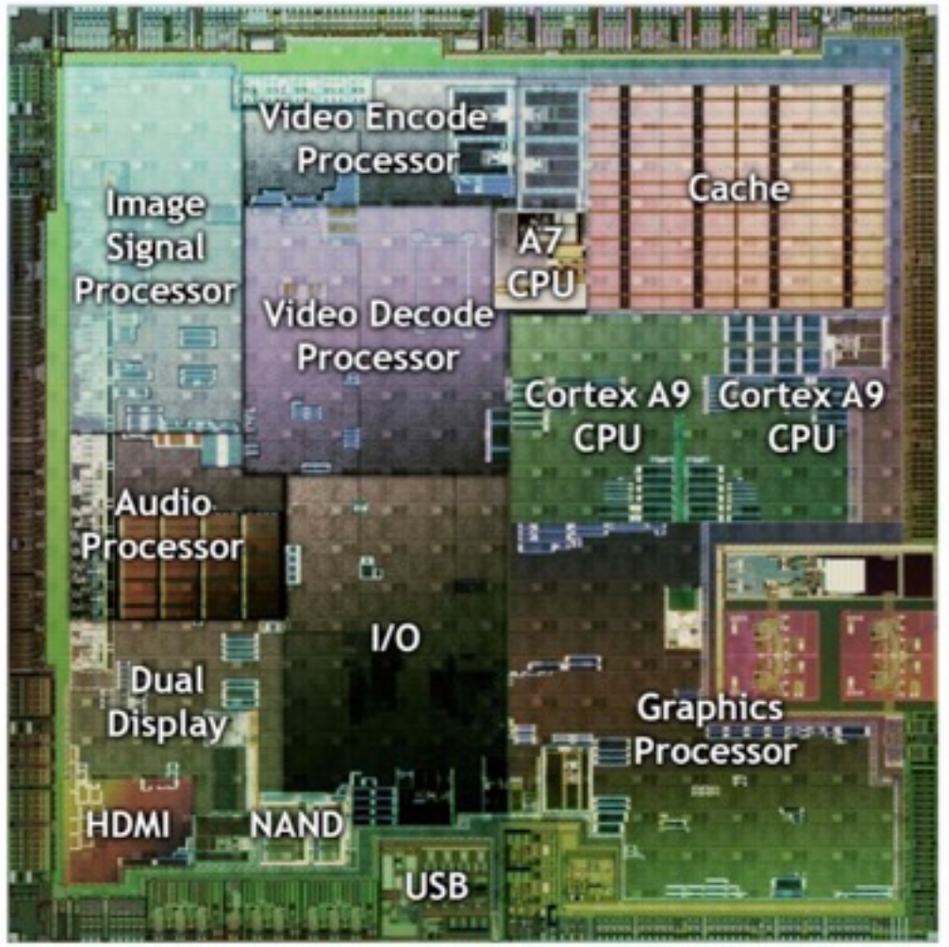
Embedded Processor: Guts



NXP



Atmega



Tegra 2

1.http://www.anandtech.com/show/4144/lg-optimus-2x-nvidia-tegra-2-review-the-first-dual-core-smartphone/3

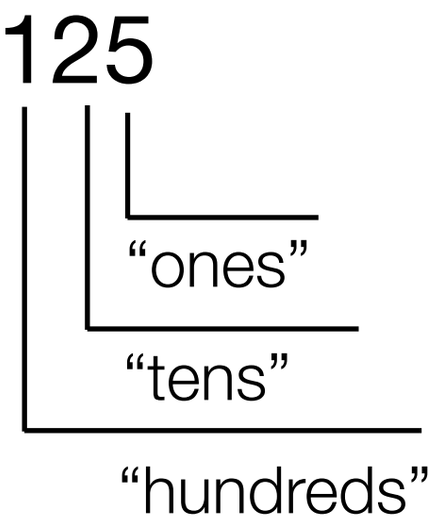
Embedded Processor: LPC1768 Block Diagram

- Core computing unit
- Memory stack
- Clocks & timers
- Digital I/O
- Analog I/O
- Communication
  - USB
  - Serial
  - Ethernet
- System drivers
  - Touch screens
  - audio
    - ⋮

Number Systems

Binary Number Systems: *a primer*

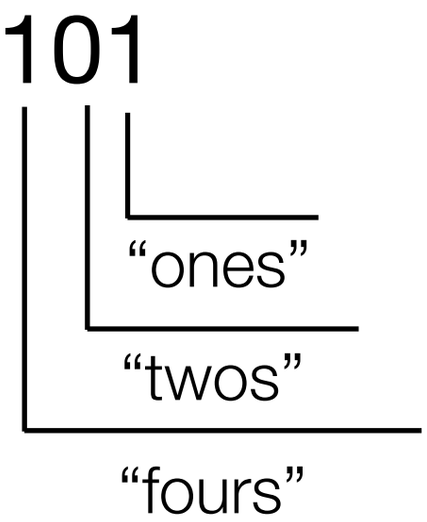
- What do we mean when we say we use a “Decimal” number system?



$$1(10^2) + 2(10^1) + 5(10^0) = 100 + 20 + 5$$

Binary Number Systems: *a primer*

- What do we mean when we refer to a “Binary” number system?



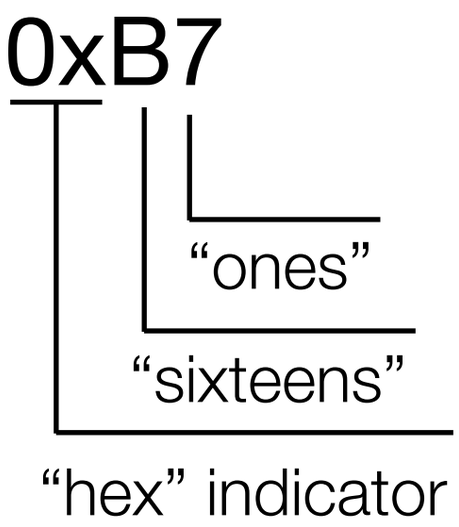
$$1(2^2) + 0(2^1) + 1(2^0) = 4 + 0 + 1$$

Any base can work:

- Base 8: “octal”
- Base 16: “hexadecimal”

Binary Number Systems: *a primer*

- What do we mean when we refer to a “Hexadecimal” number system?



$$11(16^1) + 7(16^0) = 176 + 7 = 183$$

Digits after 0…9 are lettered A…F

Bits, Bytes, & Binary

- A “bit” is just a “binary digit” — either a 0 or a 1
- A “Byte” is a string of 8 “bits,” — 01011010 etc., with 256 combinations
- How many Bytes is a kilobyte (1 kB)?

  Answer: 1024, or 2^{10}. A megabyte is 2^{20} and a gigabyte is 2^{30}.

“There are 10 kinds of people in the world: those who understand binary, and those who don’t.”

Storing Numbers in a Computer

- Storing an integer
  - Signed integers? How many bits?
- Storing decimal numbers
  - Fixed point decimals
  - Floating point and dynamic range
- How does a computer store a String (of characters)?

Communicating Digital Information

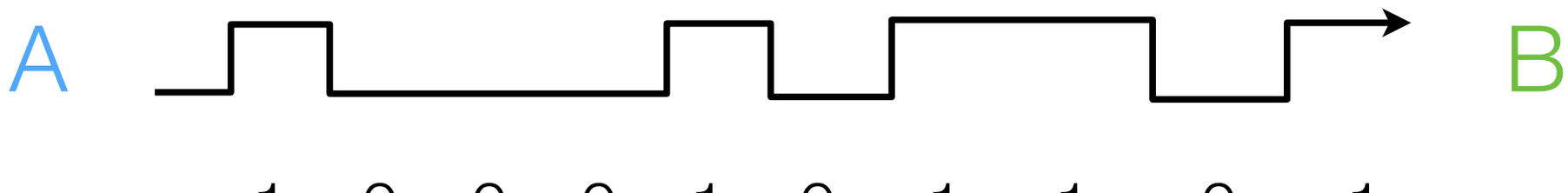
Communication

- Embedded Systems are composed of many parts, often physically separated
- How can we communicate information from one “subsystem” to another?
  - With wires? Without wires (another lecture)?
- What are some of the design considerations when communicating between two computers?

Wired Digital Communication

- (Embedded) computer A wants to tell computer B what temperature it is. How might it do this using a few wires?

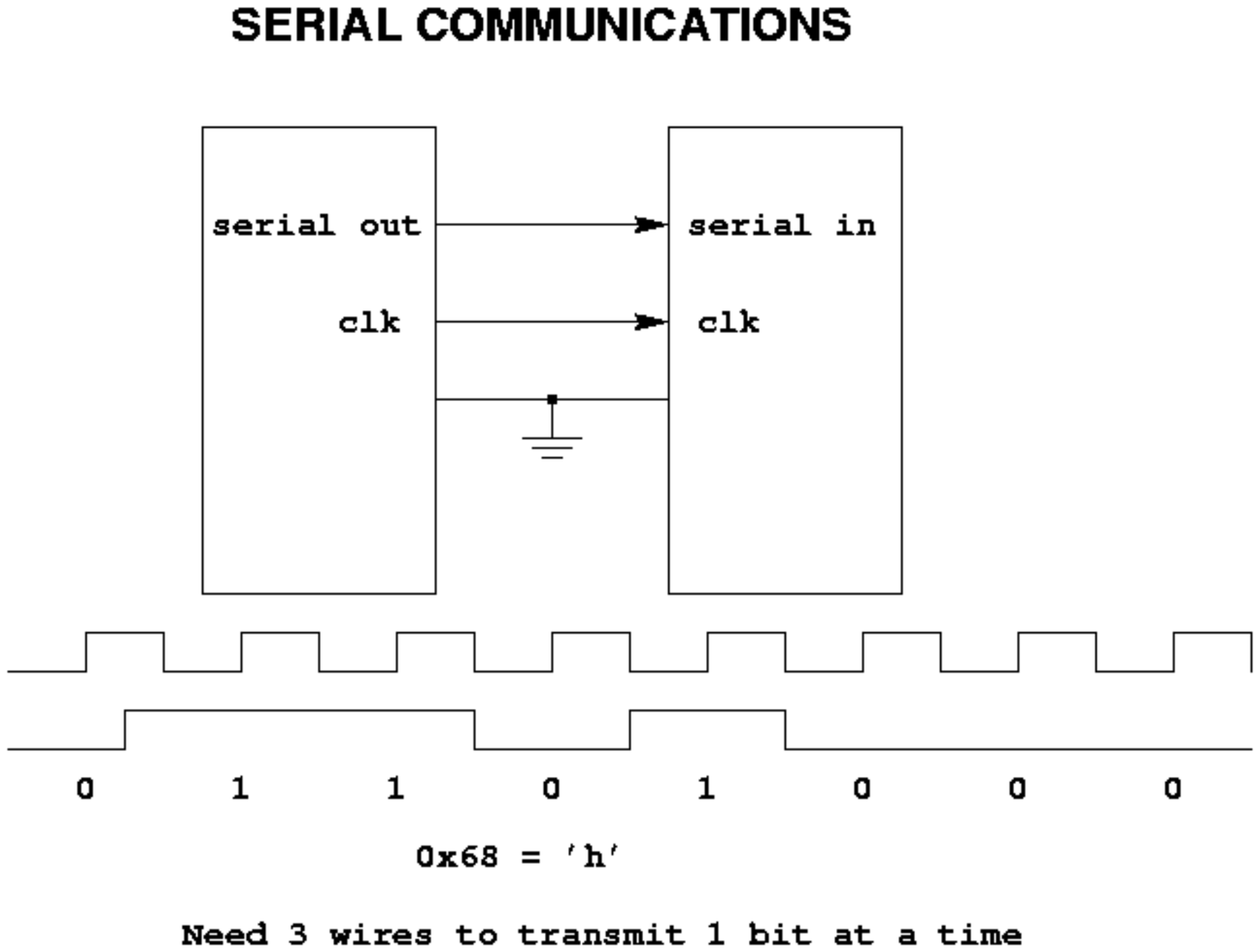
  One way:



  B must sample this wire at even intervals, at a rate that A and B agree upon. This is called “serial” communication — data is communicated serially.

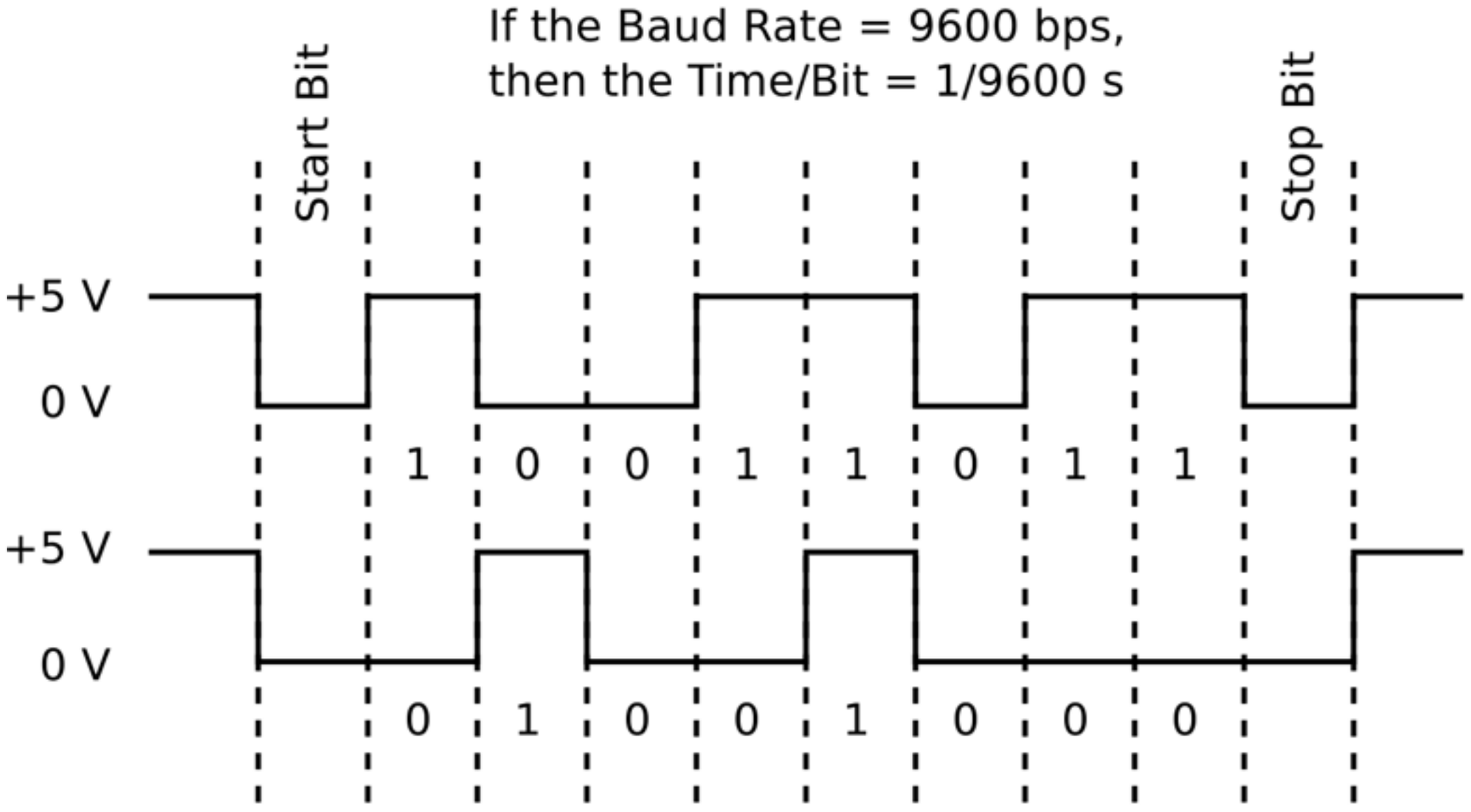
  For more information, look up UART, SPI, I^2C, & CAN protocols.

Synchronous Communication



http://www.ee.nmt.edu/~rison/ee308_spr99/lectures.html

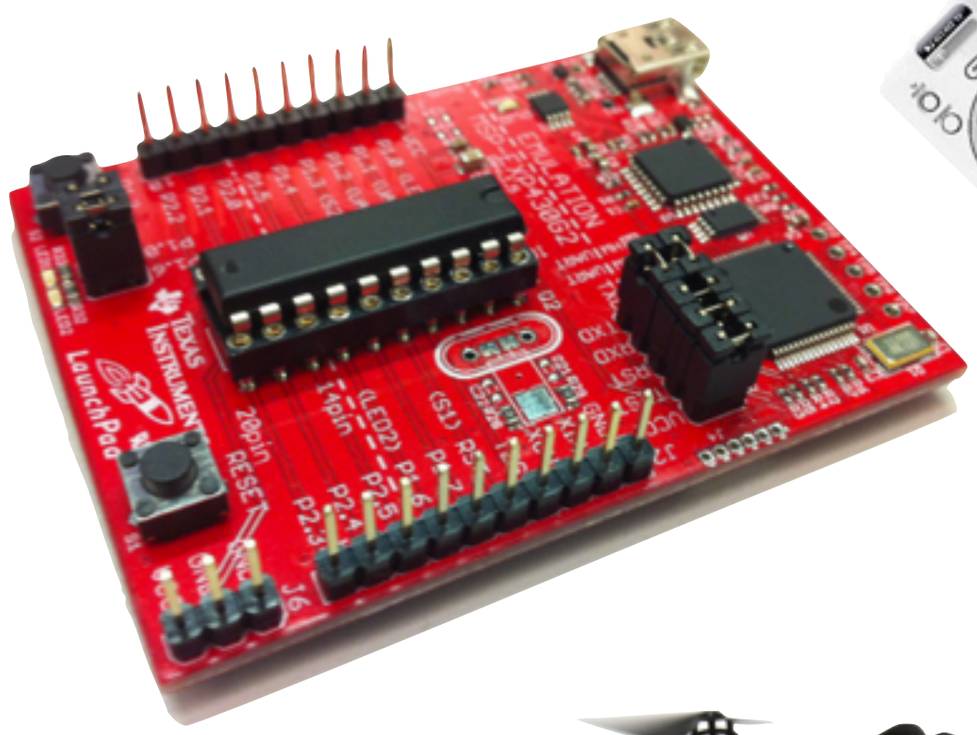
Serial “Word”

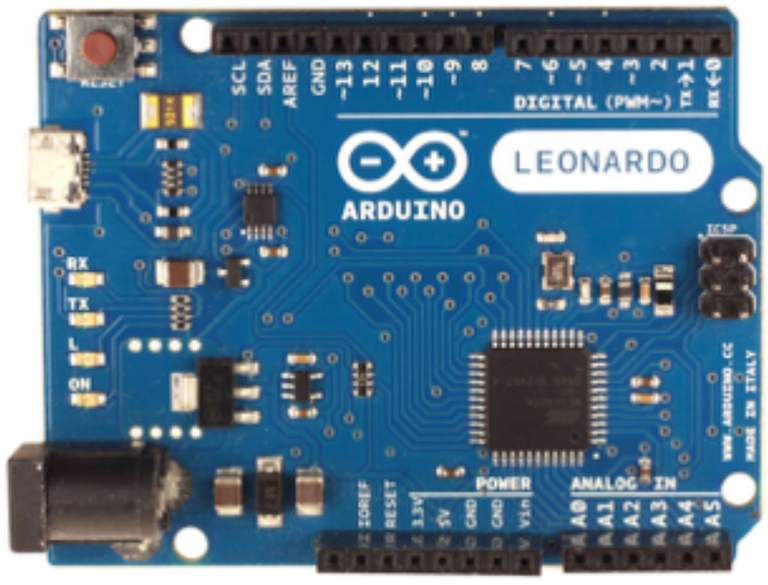

If the Baud Rate = 9600 bps,
then the Time/Bit = 1/9600 s
Start Bit
Stop Bit
+5 V
0 V
1 0 0 1 1 0 1 1
+5 V
0 V
0 1 0 0 1 0 0 0

Embedded Processor: LPC1768 Block Diagram

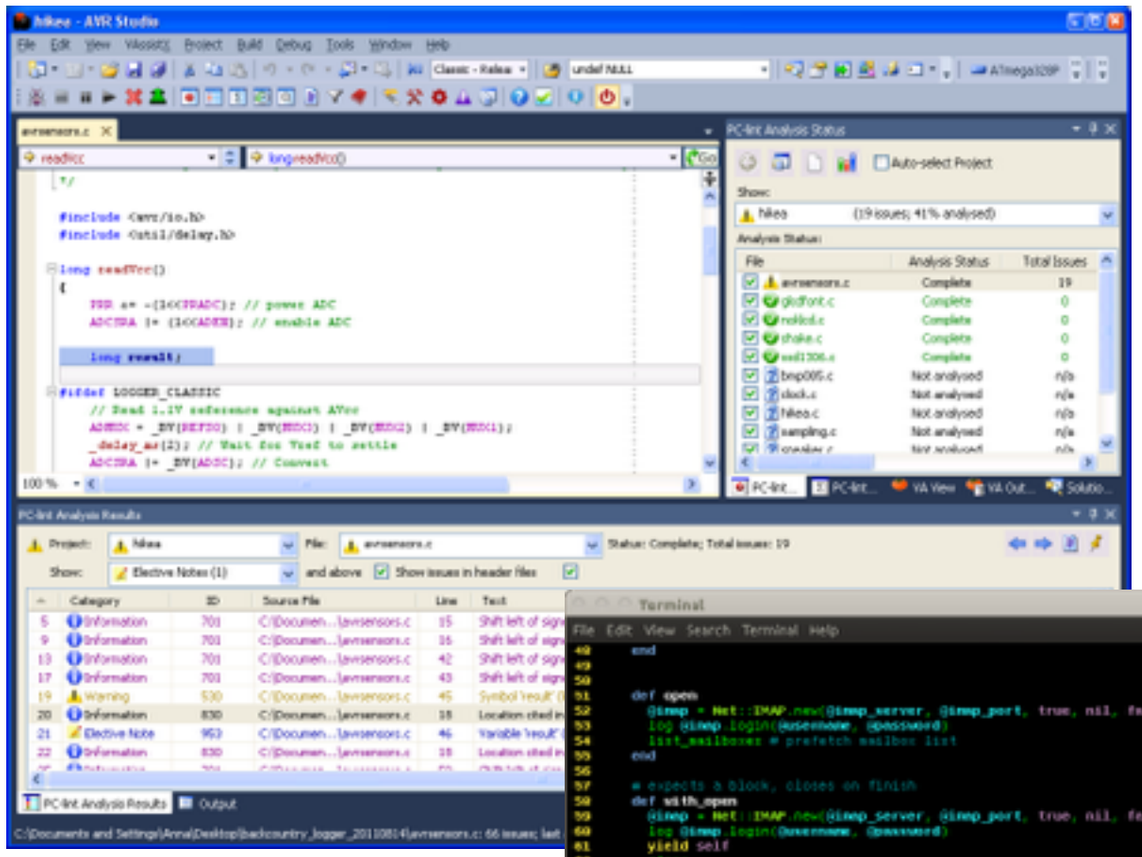
How do I develop an Embedded System?

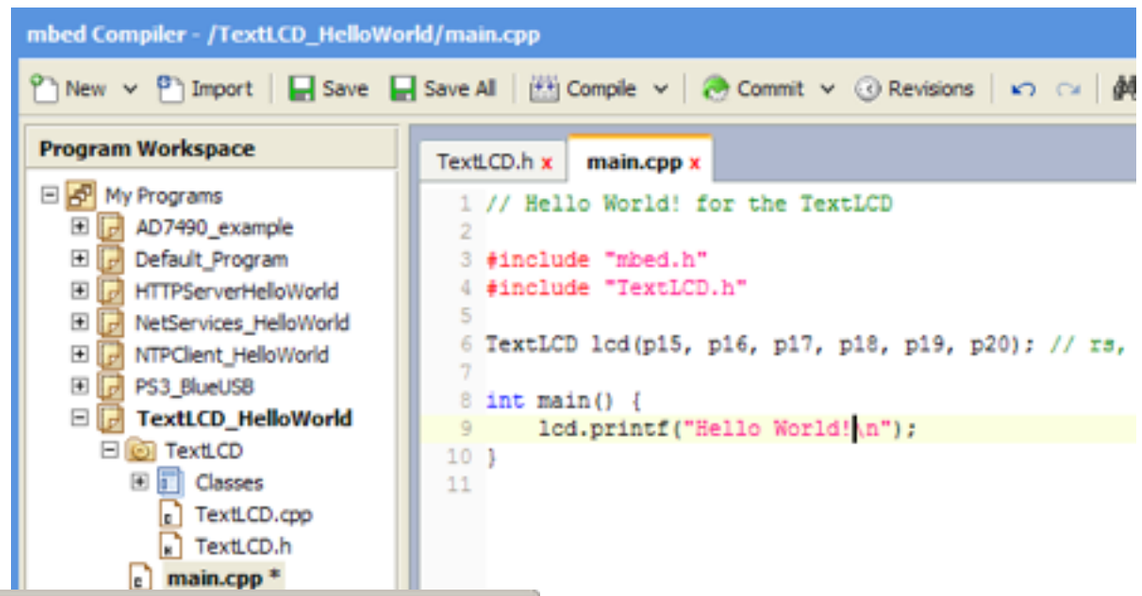
Typical *Embedded* Platforms

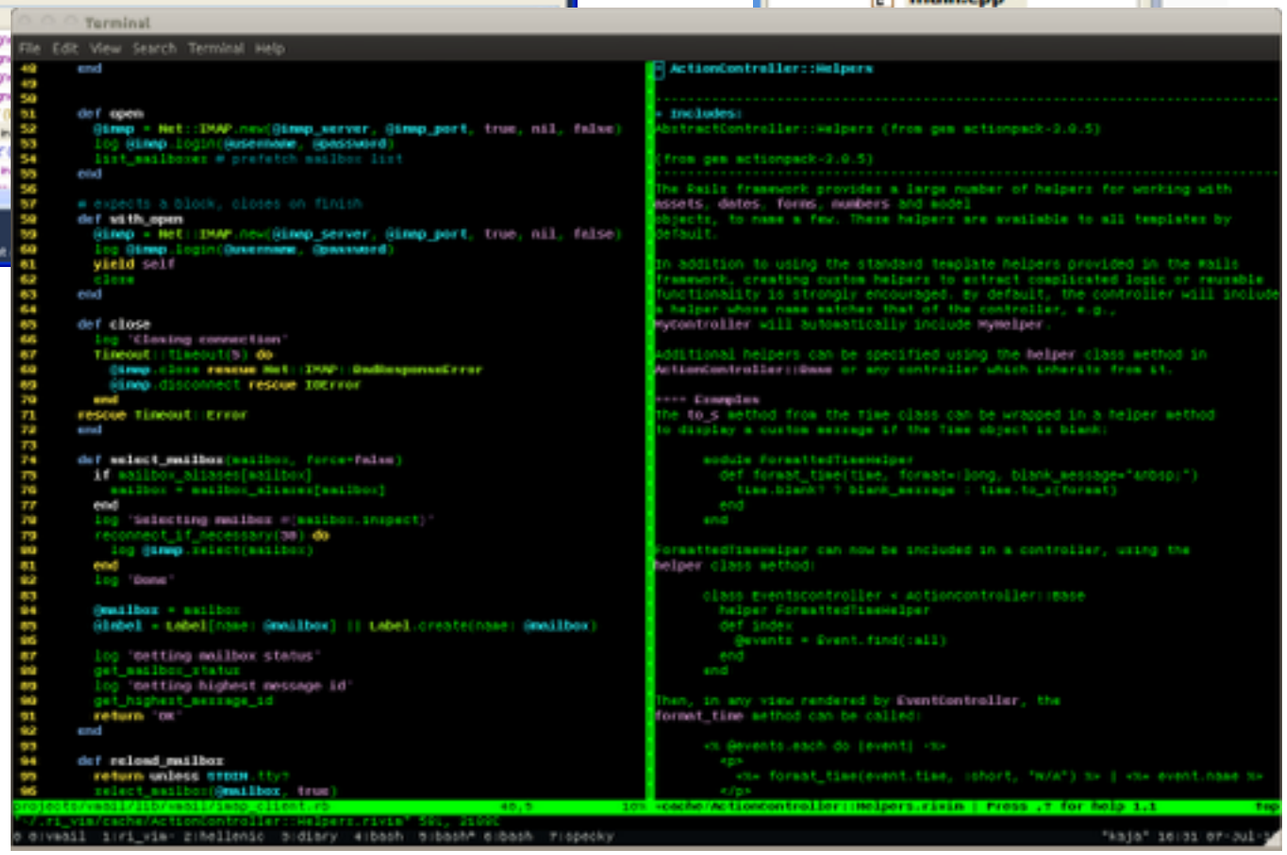




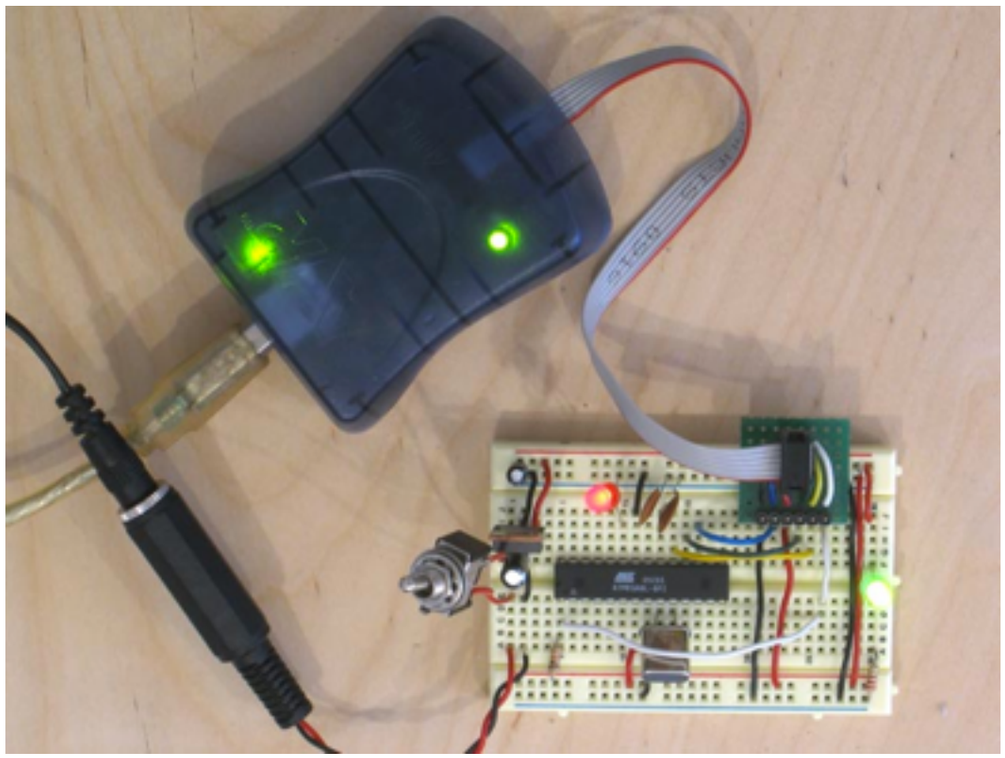
Typical Development Environments





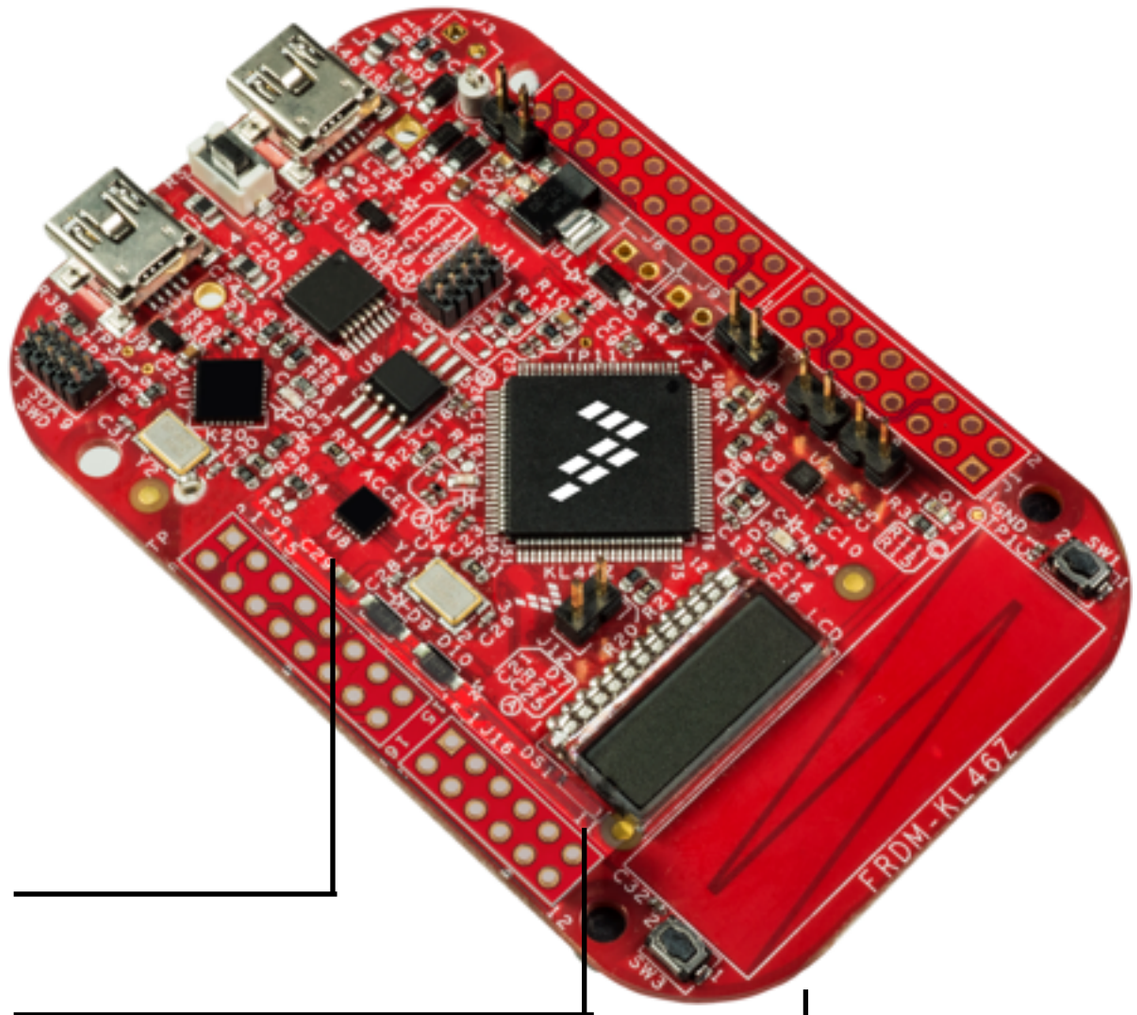


Typical Development Environments



Your Embedded Platform — “mbed”

- Online IDE
- Compiles Online
- C++
- Tons of libraries
- Fast prototyping platform
- Drag and drop programming



Accelerometer

LCD Display

Capacitive Touch

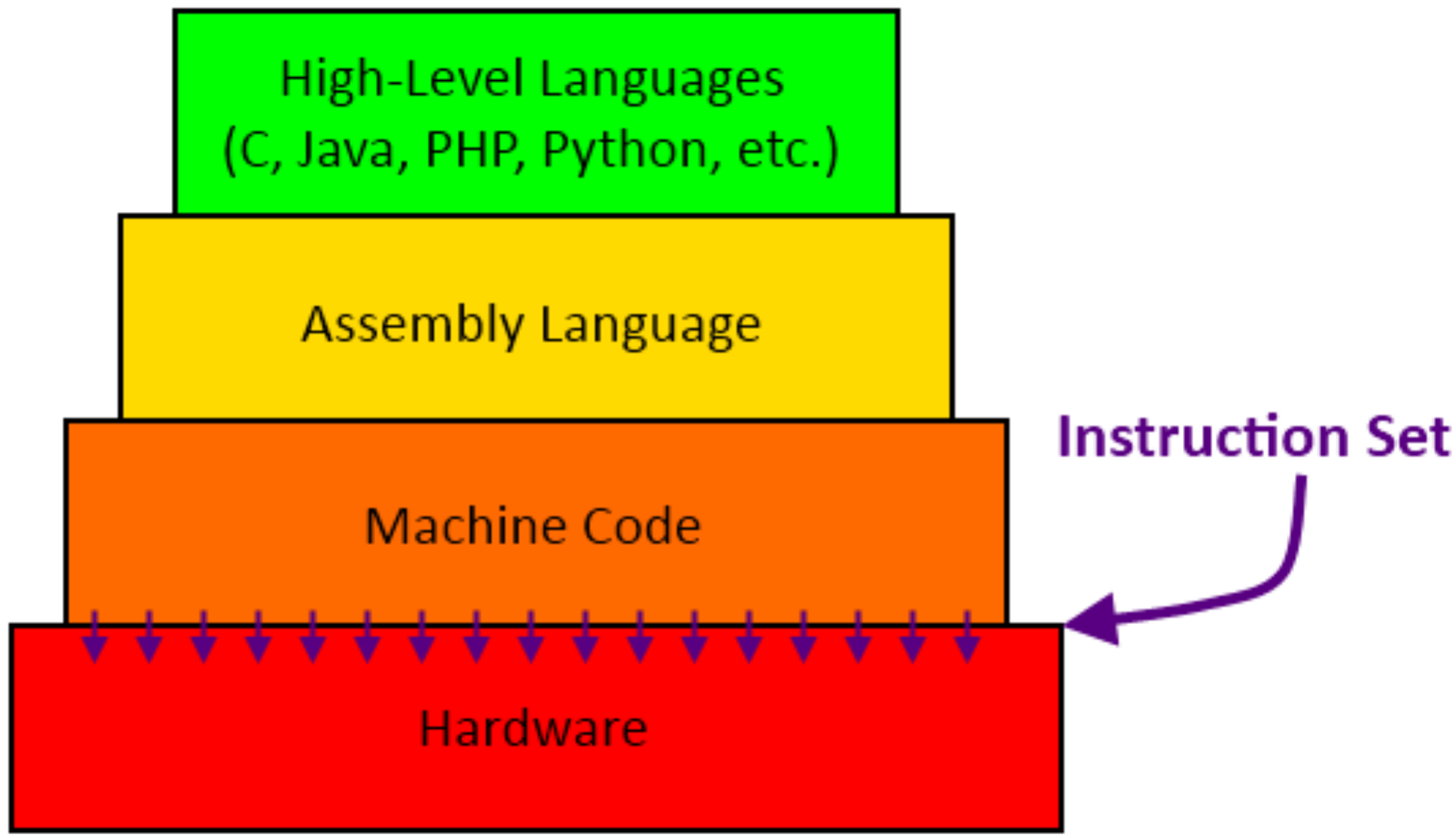
Programming an Embedded Device

- Program code is closely tied to physical hardware, so we usually program in

  C language

  and occasionally

  C++, Java, Lua



Introduction to C and C++

Primitives and Operators

Data types

- integers - int
  - int x = 1;
- floating point & double
  - float x = 0.23;
  - double x = 1.55;
- characters - char
  - char x = ‘c’;
- multiple characters - string
  - string x = “hello world”;

Operators

- assignment =
- arithmetic +,-,*,/,%
- relational <, >, <=, >=, !=, ==
- logic !, &&, ||

Comments

- //, /* ... */

Arrays

- int myArray[3] = { 5, 5, 5 };
- int a[2][2] = { {0, 1} ,{2,3} };
- Accessed like python

C: Control Structure

```
if ( a > b ) {
   ...
} else {
   ...
}
```

C: Control Structure

```
while ( a != b ) {
   ...
}
```

C: Control Structure

```
for (int i = 0; i < 100; i++) {
   ...
}
```

```
for (initialize; end condition; update) {
   ...
}
```

C: Functions

```
int addition (int a, int b) {
    int results;
    results = a + b;
    return results;
}
```

Return Types

- int, double, char, void, …

C: Array Operations

```
float sensorvals[10];
for(int i=0; i<10; i++){
        sensorvals[i] = i*i;
}
```

```
float mean = 0.0;
for(int i=0; i<10; i++){
        mean += sensorvals[i]/10.0;
}
```

C++: Objects (Classes)

```
// classes example
class Rectangle {
    int width, height;
  public:
    void set_values (int,int);
    int area() {return width*height;}
};

void Rectangle::set_values (int x, int y) {
  width = x;
  height = y;
}

int main () {
    Rectangle rect;
    rect.set_values (3,4);
    printf("area: %d", rect.area() );
    return 0;
}
```

Object name

Object variables

Object functions

Defining object functions outside of a class

Initializing an object

Calling an object function

C++ Compilation

Human Readable Language

↓

Intermediate Language

↓

Machine Code

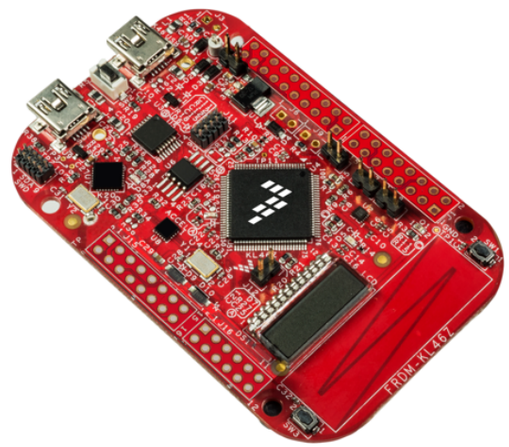
C++

↓

Assembly

↓

1001010101



C++ vs Python

- C++ programs run faster
- Compiler optimizes code for processor
- C++ requires about 4 times more lines of code
- Python works better as a “glue” language
- Python is a good rapid prototyping language

Welcome to Embedded Programming!

Your Embedded Programming Task

- Use an mbed to design a remote controller for a 2 player video game! The better your controller, the better you'll be able to play the game. The winning team will be rewarded! *(mbed?)*

- ***Steps:***
    1. Become familiar with mbed programming, C/C++, and reading the various sensors on the mbed
    2. Be able to perform gesture recognition using the on-board 3D accel.
    3. Be able to communicate with your wireless radio
    4. Create your wireless controller combining steps 2 and 3
    5. Compete!

mBed Account

- Sign up for mbed account @ www.mbed.org
- Your device information @ http://mbed.org/platforms/FRDM-KL46Z/

C++ Program

mbed Compiler - /HelloWorld3/main.cpp

New | Import | Workspace | Save | Save All | Compile | Commit | Revisions

Program Workspace

- My Programs
  - blinker
  - ee202a_hw1
  - ee202a_hw3_Part_1
  - ee202a_hw3_Part_2_1
  - ee202a_hw3_Part_2_2
  - ee202a_hw3_part_2_3
  - ee202a_hw4_jason
  - ee202a_hw4_newton_v1
  - ee202a_hw4_newton_v2
  - ee202a_hw6_masterCOM
  - ee202a_hw6_slaveCOM
  - FountainTest
  - HelloCosm
  - HelloWorld

main.cpp x

```
1  #include "mbed.h"
2
3  DigitalOut myled(LED1);
4
5  int main() {
6
7
8
9
10 }
11
```

library

Compiler Output for Program: HelloWorld3

Description | ErrNo | Resource

C++ Program

mbed Compiler - /HelloWorld3/main.cpp

New | Import | Workspace | Save | Save All | Compile | Commit | Revisions

Program Workspace

- My Programs
  - blinker
  - ee202a_hw1
  - ee202a_hw3_Part_1
  - ee202a_hw3_Part_2_1
  - ee202a_hw3_Part_2_2
  - ee202a_hw3_part_2_3
  - ee202a_hw4_jason
  - ee202a_hw4_newton_v1
  - ee202a_hw4_newton_v2
  - ee202a_hw6_masterCOM
  - ee202a_hw6_slaveCOM
  - FountainTest
  - HelloCosm
  - HelloWorld

main.cpp x

```
1  #include "mbed.h"
2
3  DigitalOut myled(LED1);
4
5  int main() {
6
7
8
9
10 }
11
```

name of “library”

directive to include said “library”

Compiler Output for Program: HelloWorld3

Description	ErrNo	Resource

C++ Program

mbed Compiler - /HelloWorld3/main.cpp

New | Import | Workspace | Save | Save All | Compile | Commit | Revisions

Program Workspace

- My Programs
  - blinker
  - ee202a_hw1
  - ee202a_hw3_Part_1
  - ee202a_hw3_Part_2_1
  - ee202a_hw3_Part_2_2
  - ee202a_hw3_part_2_3
  - ee202a_hw4_jason
  - ee202a_hw4_newton_v1
  - ee202a_hw4_newton_v2
  - ee202a_hw6_masterCOM
  - ee202a_hw6_slaveCOM
  - FountainTest
  - HelloCosm
  - HelloWorld

main.cpp x

```
1  #include "mbed.h"
2
3  DigitalOut myled(LED1);
4
5  int main() {
6
7
8
9
10 }
11
```

Create a digital pin object

Compiler Output for Program: HelloWorld3

Description	ErrNo	Resource

C++ Program

mbed Compiler - /HelloWorld3/main.cpp

New | Import | Workspace | Save | Save All | Compile | Commit | Revisions

Program Workspace

- My Programs
  - blinker
  - ee202a_hw1
  - ee202a_hw3_Part_1
  - ee202a_hw3_Part_2_1
  - ee202a_hw3_Part_2_2
  - ee202a_hw3_part_2_3
  - ee202a_hw4_jason
  - ee202a_hw4_newton_v1
  - ee202a_hw4_newton_v2
  - ee202a_hw6_masterCOM
  - ee202a_hw6_slaveCOM
  - FountainTest
  - HelloCosm
  - HelloWorld

main.cpp x

```
1  #include "mbed.h"
2
3  DigitalOut myled(LED1);
4
5  int main() {
6
7
8
9
10 }
11
```

set specific pin

variable name

Object Name

Compiler Output for Program: HelloWorld3

Description	ErrNo	Resource

C++ Program

mbed Compiler - /HelloWorld3/main.cpp

New | Import | Workspace | Save | Save All | Compile | Commit | Revisions

Program Workspace

- My Programs
  - blinker
  - ee202a_hw1
  - ee202a_hw3_Part_1
  - ee202a_hw3_Part_2_1
  - ee202a_hw3_Part_2_2
  - ee202a_hw3_part_2_3
  - ee202a_hw4_jason
  - ee202a_hw4_newton_v1
  - ee202a_hw4_newton_v2
  - ee202a_hw6_masterCOM
  - ee202a_hw6_slaveCOM
  - FountainTest
  - HelloCosm
  - HelloWorld

main.cpp x

```
1  #include "mbed.h"
2
3  DigitalOut myled(LED1);
4
5  int main() {
6
7
8
9
10 }
11
```

Main function

Compiler Output for Program: HelloWorld3

Description | ErrNo | Resource

C++ Program

mbed Compiler - /HelloWorld3/main.cpp

New | Import | Workspace | Save | Save All | Compile | Commit | Revisions

Program Workspace

- My Programs
  - blinker
  - ee202a_hw1
  - ee202a_hw3_Part_1
  - ee202a_hw3_Part_2_1
  - ee202a_hw3_Part_2_2
  - ee202a_hw3_part_2_3
  - ee202a_hw4_jason
  - ee202a_hw4_newton_v1
  - ee202a_hw4_newton_v2
  - ee202a_hw6_masterCOM
  - ee202a_hw6_slaveCOM
  - FountainTest
  - HelloCosm
  - HelloWorld

main.cpp x

```
1  #include "mbed.h"
2
3  DigitalOut myled(LED1);
4
5  int main() {
6
7
8
9
10 }
11
```

parameters (none)

function name

return type (why is it an int?)

Compiler Output for Program: HelloWorld3

Description | ErrNo | Resource

C++ Program

mbed Compiler - /HelloWorld3/main.cpp

New | Import | Workspace | Save | Save All | Compile | Commit | Revisions

Program Workspace

- My Programs
  - blinker
  - ee202a_hw1
  - ee202a_hw3_Part_1
  - ee202a_hw3_Part_2_1
  - ee202a_hw3_Part_2_2
  - ee202a_hw3_part_2_3
  - ee202a_hw4_jason
  - ee202a_hw4_newton_v1
  - ee202a_hw4_newton_v2
  - ee202a_hw6_masterCOM
  - ee202a_hw6_slaveCOM
  - FountainTest
  - HelloCosm
  - HelloWorld

main.cpp x

```
1  #include "mbed.h"
2
3  DigitalOut myled(LED1);
4
5  int main() {
6
7
8
9
10 }
11
```

curly braces

Compiler Output for Program: HelloWorld3

Description | ErrNo | Resource

C++ Program

mbed Compiler - /HelloWorld3/main.cpp

New | Import | Workspace | Save | Save All | Compile | Commit | Revisions

Program Workspace

- My Programs
  - blinker
  - ee202a_hw1
  - ee202a_hw3_Part_1
  - ee202a_hw3_Part_2_1
  - ee202a_hw3_Part_2_2
  - ee202a_hw3_part_2_3
  - ee202a_hw4_jason
  - ee202a_hw4_newton_v1
  - ee202a_hw4_newton_v2
  - ee202a_hw6_masterCOM
  - ee202a_hw6_slaveCOM
  - FountainTest
  - HelloCosm
  - HelloWorld

main.cpp x

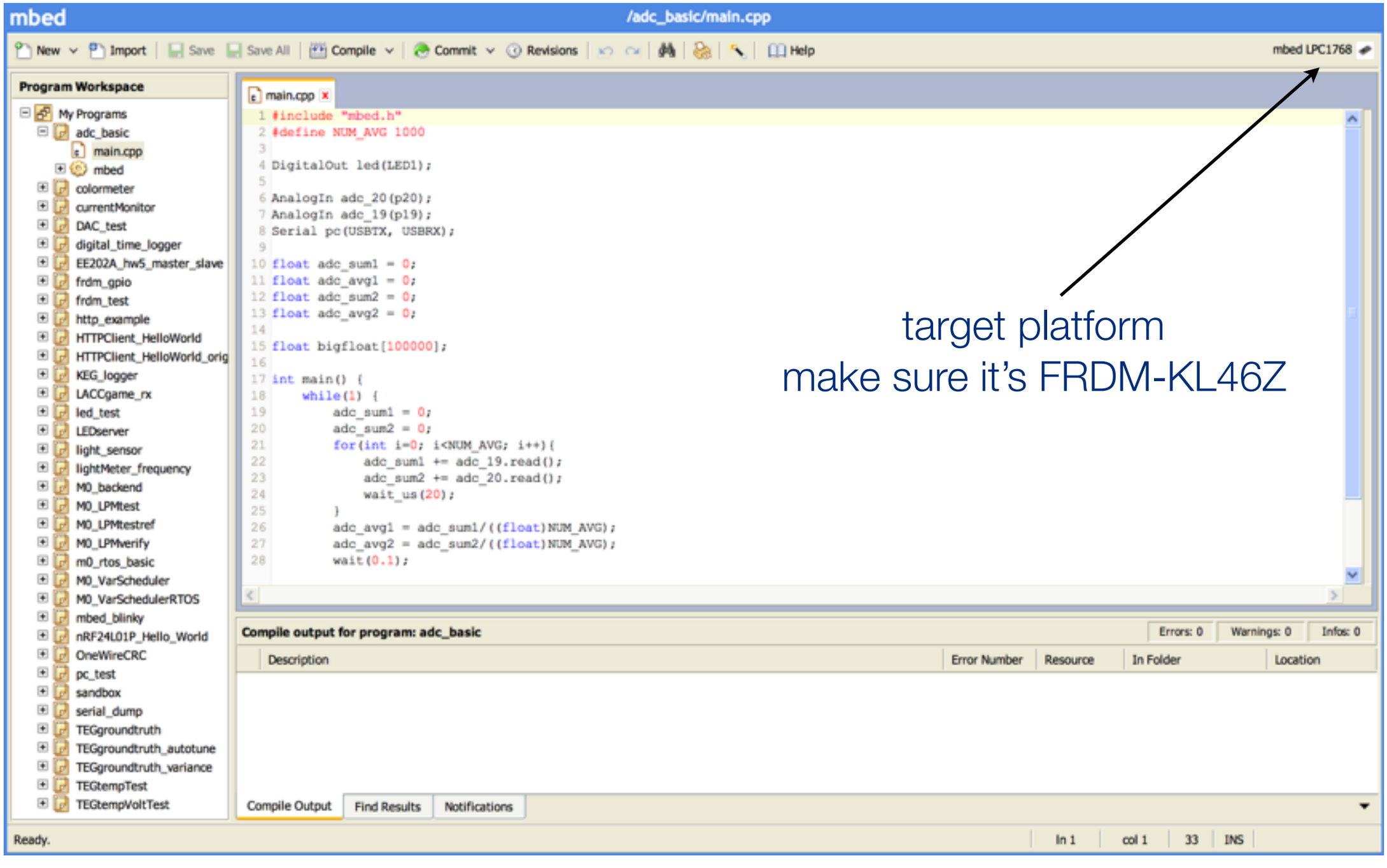
```
1  #include "mbed.h"
2
3  DigitalOut myled(LED1);
4
5  int main() {
6
7
8
9
10 }
11
```

compiler

Compiler Output for Program: HelloWorld3

Description	ErrNo	Resource

C++ Program


mbed
/adc_basic/main.cpp
New
Import
Save
Save All
Compile
Commit
Revisions
Help
mbed LPC1768
Program Workspace
My Programs
adc_basic
main.cpp
mbed
colormeter
currentMonitor
DAC_test
digital_time_logger
EE202A_hw5_master_slave
frdm_gpio
frdm_test
http_example
HTTPClient_HelloWorld
HTTPClient_HelloWorld_orig
KEG_logger
LACCgame_rx
led_test
LEDserver
light_sensor
lightMeter_frequency
M0_backend
M0_LPMtest
M0_LPMtestref
M0_LPMverify
m0_rtos_basic
M0_VarScheduler
M0_VarSchedulerRTOS
mbed_blinky
nRF24L01P_Hello_World
OneWireCRC
pc_test
sandbox
serial_dump
TEGgroundtruth
TEGgroundtruth_autotune
TEGgroundtruth_variance
TEGtempTest
TEGtempVoltTest
main.cpp
#include "mbed.h"
#define NUM_AVG 1000
DigitalOut led(LED1);
AnalogIn adc_20(p20);
AnalogIn adc_19(p19);
Serial pc(USBTX, USBRX);
float adc_sum1 = 0;
float adc_avg1 = 0;
float adc_sum2 = 0;
float adc_avg2 = 0;
float bigfloat[100000];
int main() {
while(1) {
adc_sum1 = 0;
adc_sum2 = 0;
for(int i=0; i<NUM_AVG; i++){
adc_sum1 += adc_19.read();
adc_sum2 += adc_20.read();
wait_us(20);
}
adc_avg1 = adc_sum1/((float)NUM_AVG);
adc_avg2 = adc_sum2/((float)NUM_AVG);
wait(0.1);
target platform
make sure it's FRDM-KL46Z
Compile output for program: adc_basic
Errors: 0
Warnings: 0
Infos: 0
Description
Error Number
Resource
In Folder
Location
Compile Output
Find Results
Notifications
Ready.
ln 1
col 1
33
INS

C++ Program

mbed Compiler - /HelloWorld3/main.cpp

New | Import | Workspace | Save | Save All | Compile | Commit | Revisions

Program Workspace

- My Programs
  - blinker
  - ee202a_hw1
  - ee202a_hw3_Part_1
  - ee202a_hw3_Part_2_1
  - ee202a_hw3_Part_2_2
  - ee202a_hw3_part_2_3
  - ee202a_hw4_jason
  - ee202a_hw4_newton_v1
  - ee202a_hw4_newton_v2
  - ee202a_hw6_masterCOM
  - ee202a_hw6_slaveCOM
  - FountainTest
  - HelloCosm
  - HelloWorld

main.cpp x

```
1  #include "mbed.h"
2
3  DigitalOut myled(LED1);
4
5  int main() {
6      while(1) {
7          myled = 1;
8          wait(0.2);
9          myled = 0;
10         wait(0.2);
11     }
12 }
```

while loop

set pin to output 1 or 3.3 V

pause 0.2 seconds

set pin to output 0 or 0 V

Compiler Output for Program: HelloWorld3

Description | ErrNo | Resource

Blinking LEDs

1. Sign up for mbed account
2. Modify the blinking LED program
3. Write a program to blink LEDs in series

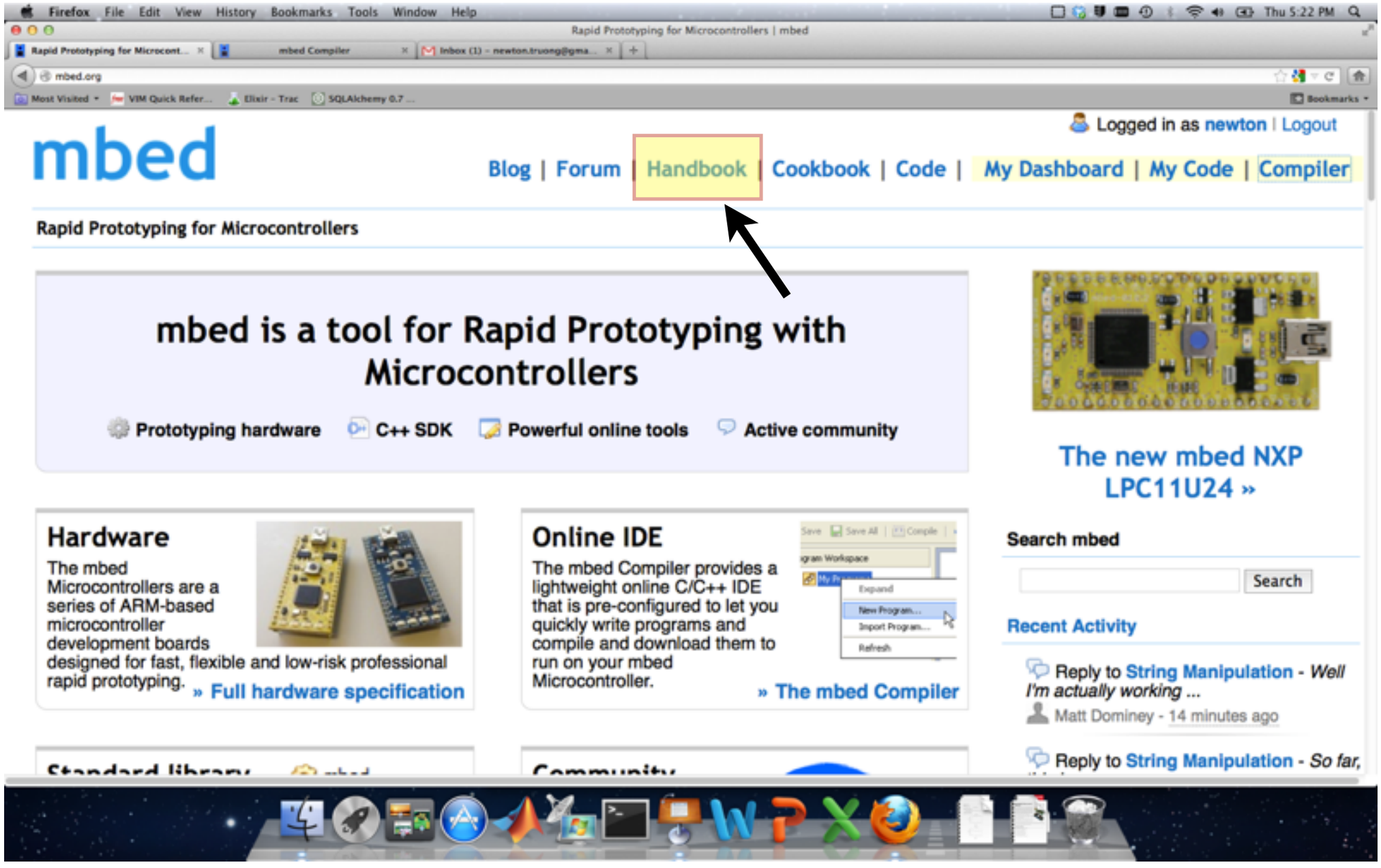
   Create new pin operations for:
   LED1, LED2, LED3, LED4

“Hello World” Terminal

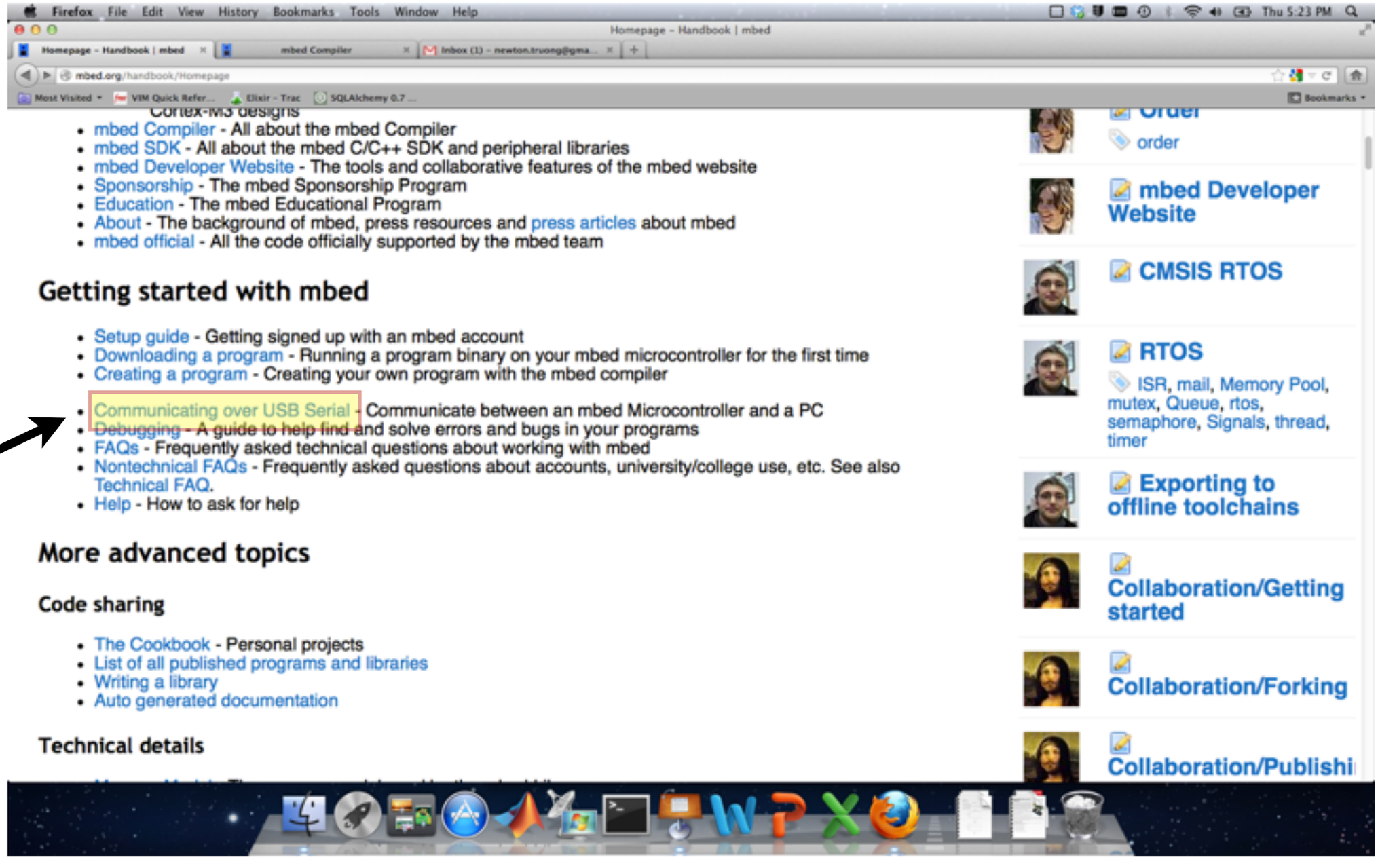
- Windows
  1. Download driver from www.mbed.org
  2. Download TeraTerm
  3. Write “Hello World” program
  4. Compile and program mbed

- MAC & Linux
  6. Write “Hello World” Program
  7. Open terminal -> screen /dev/tty.usbmodem (tab complete)
  8. Compile and program mbed

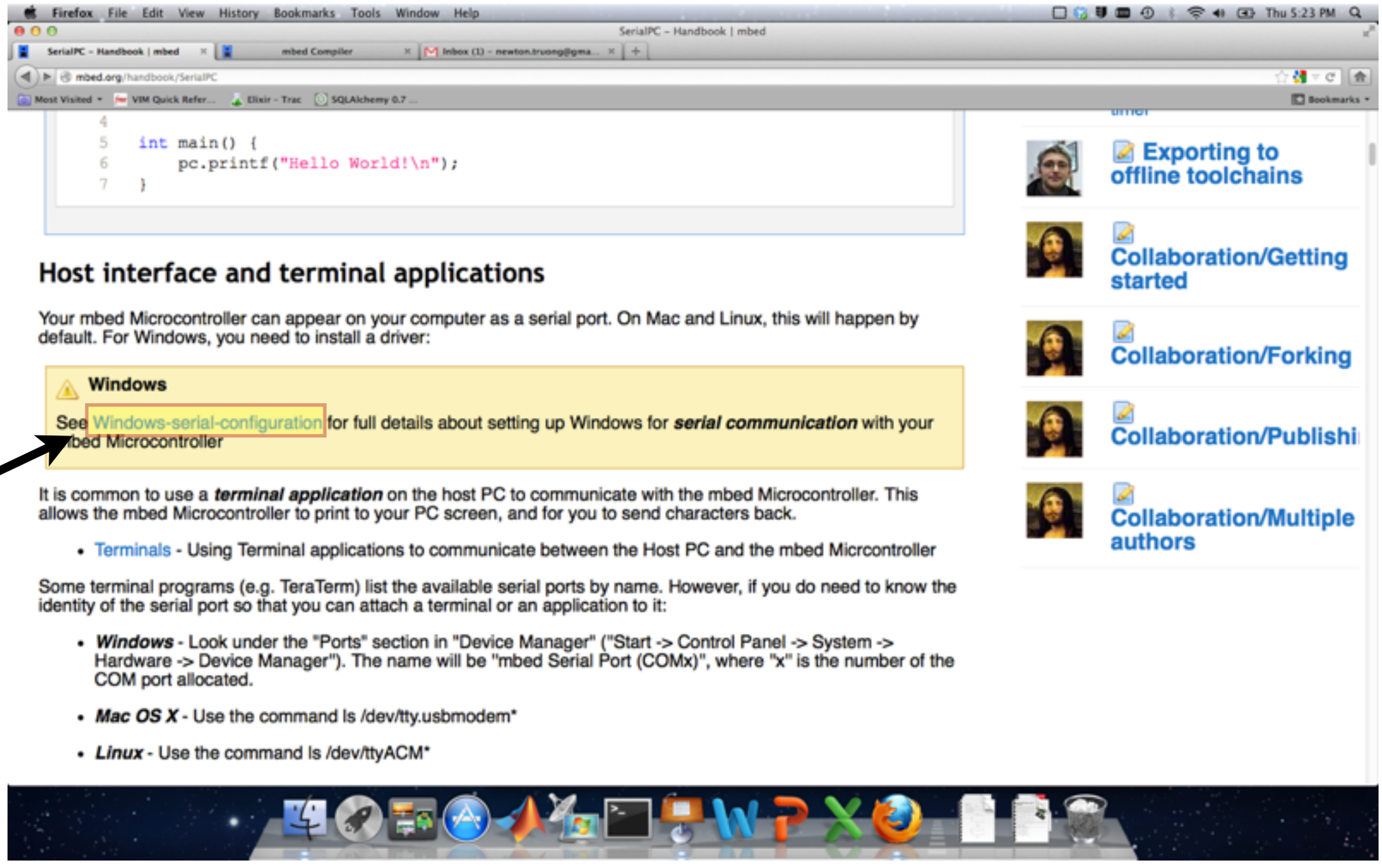
“Hello World” Terminal (Windows)



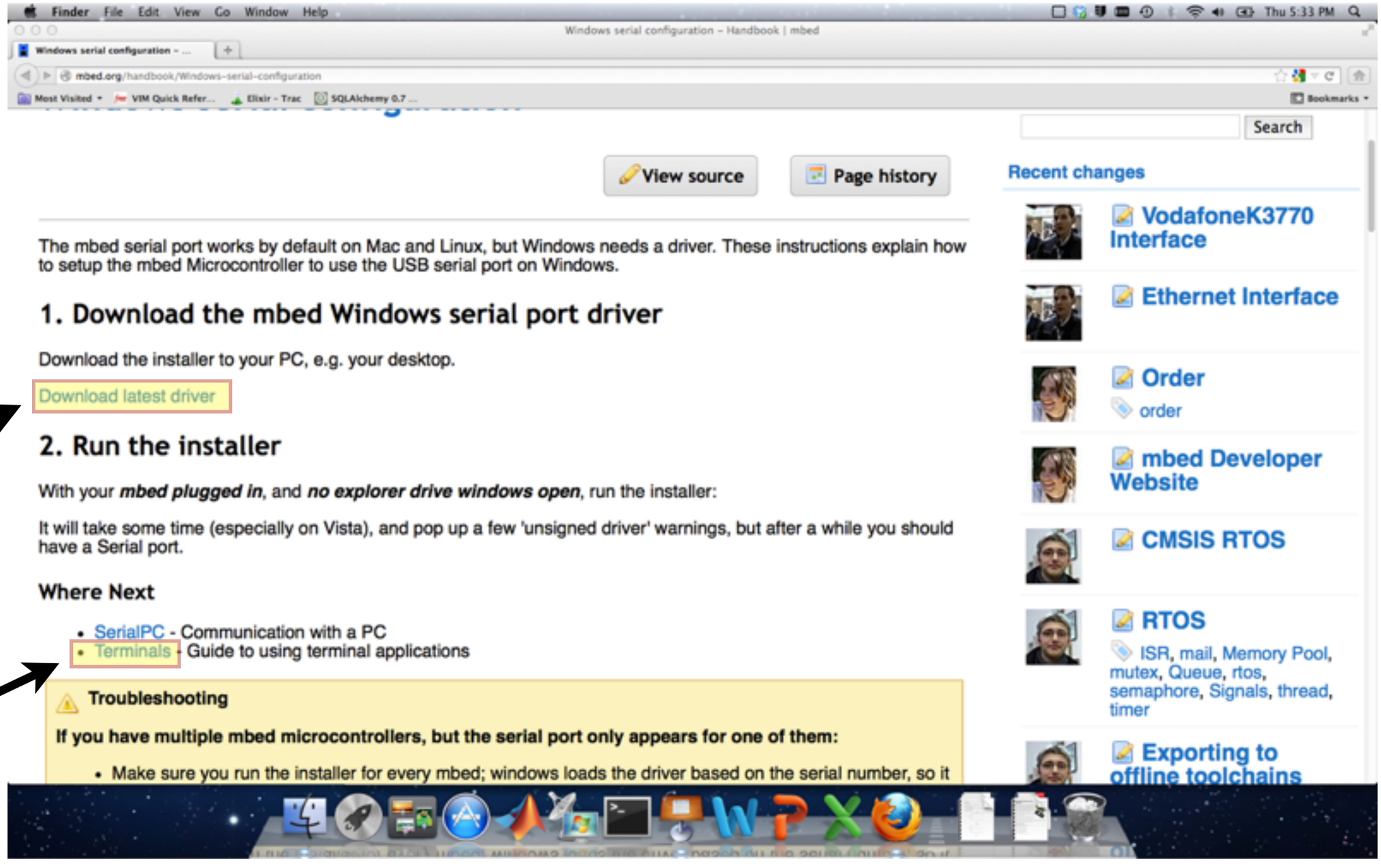
“Hello World” Terminal (Windows)



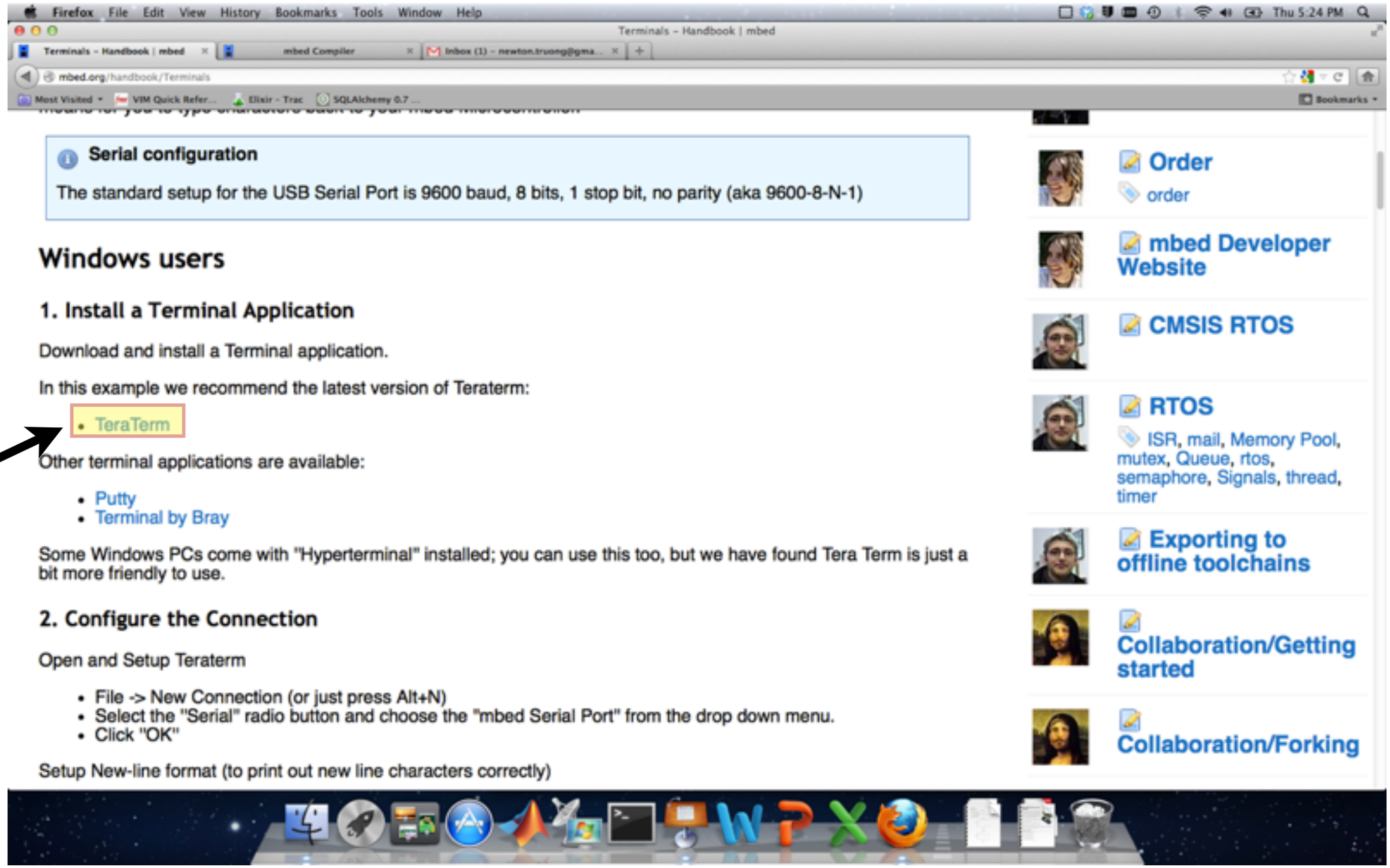
“Hello World” Terminal (Windows)



“Hello World” Terminal (Windows)



“Hello World” Terminal (Windows)



“Hello World” Program

mbed Compiler - /HelloWorld3/main.cpp

New | Import | Workspace | Save | Save All | Compile | Commit | Revisions | Format | mbed NXP LPC1768

Program Workspace

- FountainTest
- HelloCosm
- HelloWorld
- HelloWorld2
- **HelloWorld3**
  - **main.cpp ***
  - mbed
- hw6_master
- hw6_master_intr
- hw6_slave
- I2CU
- IrDA_Rx
- IrDA_Tx
- IrDA_TxStressTest
- Lights
- MSL_Potentiostat
- OpticalScanner
- runOptiTest
- ServoTest
- SolenoidCtrl
- testAngOut

main.cpp x — Expand

```
1  #include "mbed.h"
2
3  Serial pc(USBTX,USBRX); // my comments
4
5  int main() {
6      while(1) {
7          pc.printf("Hello World\r\n");
8          wait(2);
9          pc.printf("Goodbye World\r\n");
10         wait(2);
11     }
12 }
```

Compiler Output for Program: HelloWorld3 — Errors: 0 | Warnings: 0 | Infos: 0

Description	ErrNo	Resource	In Folder	Location

Ready. INS

“Hello World” Terminal (Mac)

- Windows
  1. Open TeraTerm
  2. Connect to mbed

- MAC
  3.Open terminal -> screen /dev/tty.usbmodem (tab complete)

Reading Sensors

- The FRDM-KL46Z (mbed) has several built-in sensors:
  - 3D accelerometer (measures accelerations)
  - Capacitive touch sensor

- The mbed API very nicely wraps the reading of these sensors into easy-to-use functions:

```
TSIAnalogSlider touchsensor(ELEC0, ELEC1, 40);

float touchval = touchsensor.readPercentage();
```

About printf — formatted printing

```
int i = 3;
pc.printf("The value of i is %d\n", i);
```

```
console >> The value of i is 3
```

```
%d - integers
%f - floats
%c - character
%s - string of characters
```

```
int num_dogs = 3;
float num_cats = 2.5;
```

```
pc.printf("There are %d dogs and %f cats\n", num_dogs, num_cats);
```

Putting it All Together

- After looking through mbed docs, you should be able to:
  - Read the touch sensor
  - Read all 3 axes of the accelerometer
  - Print sensor readings over USB and display on a computer

- Now we need to:
  - Detect “gesture” using your accelerometer
  - Communicate commands wirelessly to play a game

Wireless Communication

- You each will be given an nRF24L01+ wireless module—a 2.4 ghz transceiver (both RX & TX) that communicates using wired serial communication (called SPI)
- There is a preexisting mbed library for the nRF24L01+ which you should find and use in your library. This will include functions for initializing your wireless module, such as this:

```
// initialize wireless comm.
my_nrf24l01p.powerUp();
my_nrf24l01p.setTransferSize( TRANSFER_SIZE );
my_nrf24l01p.setReceiveMode();
my_nrf24l01p.enable();
my_nrf24l01p.setAirDataRate(2000);
my_nrf24l01p.setRxAddress(0xE7E7E7E7E7);
my_nrf24l01p.setTxAddress(0xCDABCDABCD);
```

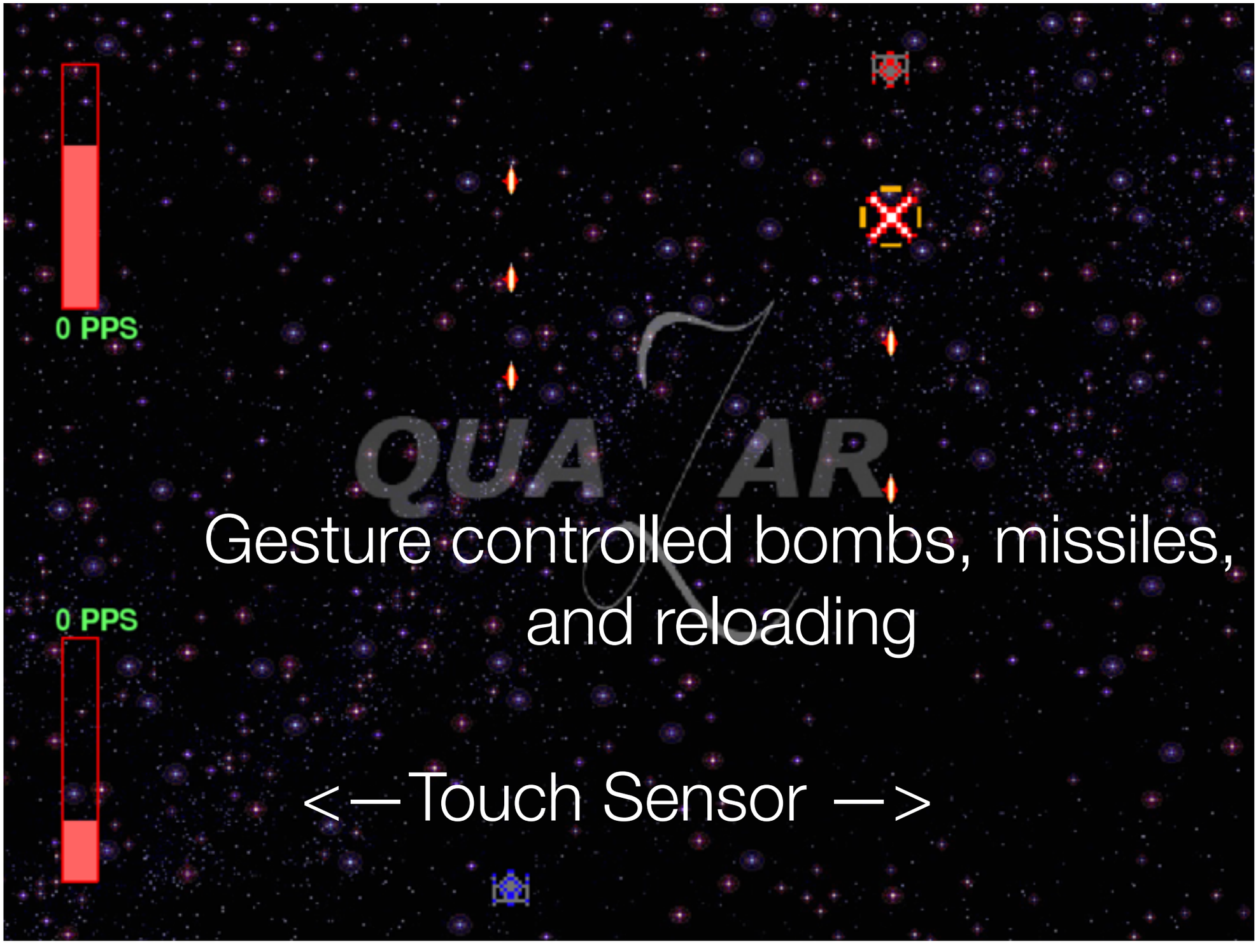
Wireless Communication

- Follow instructions online for connecting your wireless module to your mbed using the provided wires, and ask for assistance if necessary.
- Follow example code to ensure that you can properly read from your radio
- Use example code to test both RX and TX with another team’s radio!

Quazar!

Quazar, the LACC-original Python Game

Quazar, the LACC-original Python Game

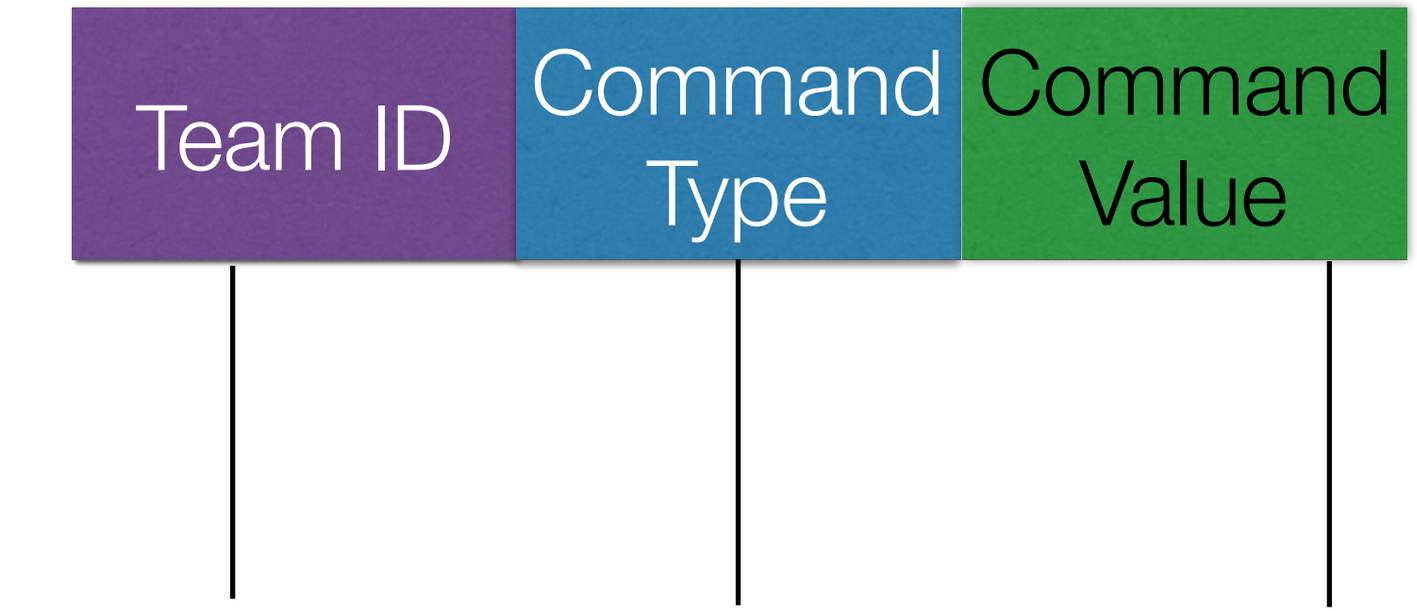


Quazar, the LACC-original Python Game

DEMO!

The Rules of Engagement

- Each team is given a unique 1-byte (0-255) identifier
- To control your player, you must wirelessly transmit a packet to the address “CDABCDABCD” at 2.402 ghz and 2000 kbps formatted like so:



The Rules of Engagement

- You start with 15 missiles, 3 bombs, and 12 HP
  - missile: 1 damage
  - bomb: 3 damage
- You must reload for more ammo (roughly 1-2 seconds)
- Your ship is out in the far reaches of space, and you can only talk to it infrequently! If you transmit more than 100 packets per second, your ship will freeze for an entire second as a penalty!

Final Words of Wisdom

- Your gestures should be recognized quickly by mbed, but with low false positives—look into filtering your data
- You'll need to limit your wireless data rate—the easiest way to do this is by adding delays in your infinite loops
- You can add smarts! Instead of one gesture shooting one missile, why not make one gesture shoot a barrage of 5 missiles?
- Don't script your entire controller—if you can't demonstrate full control over your vehicle (i.e. your controller is automated), you'll be disqualified!

Further Research Topics

- If you’re curious and you’d like to learn more about embedded computing, check out these cool topics:

  - Real Time Operating Systems (RTOS) and real time scheduling
  - Time Synchronization
  - Wired communication protocols (I2C, SPI, CAN, UART/USART)
  - Low power embedded computing practices (duty cycling, interrupts, low power modes, processor frequencies). TI MSP430 Processors
  - RISC architectures and ARM processors — Cortex M0-M4
  - Specialized hardware accelerators—DSP and FPGA processors.

See You on the Battlefield

- Program and test today
- You'll do battle tomorrow, winners will walk away with something special!
- Happy coding :)