Internet of Things & Embedded Systems

Akash Deep Singh

Networked & Embedded Systems Lab

July 29-30, 2019

Note: modified from original LACC 2016 slides by Paul Martin

Course structure

- Learning by doing:
  - 8 mini-projects/activities
- Think of this module as a discussion rather than a lecture.
  - Ask me questions
  - I will ask you questions
- We will be watching a lot of cool videos

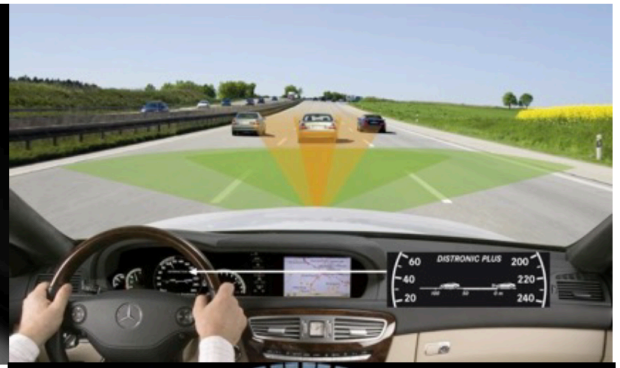
What are Embedded Systems?

- Wikipedia: *An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints.*
- For the purposes of this course: General-purpose computers: flexible, wide range of needs.

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/

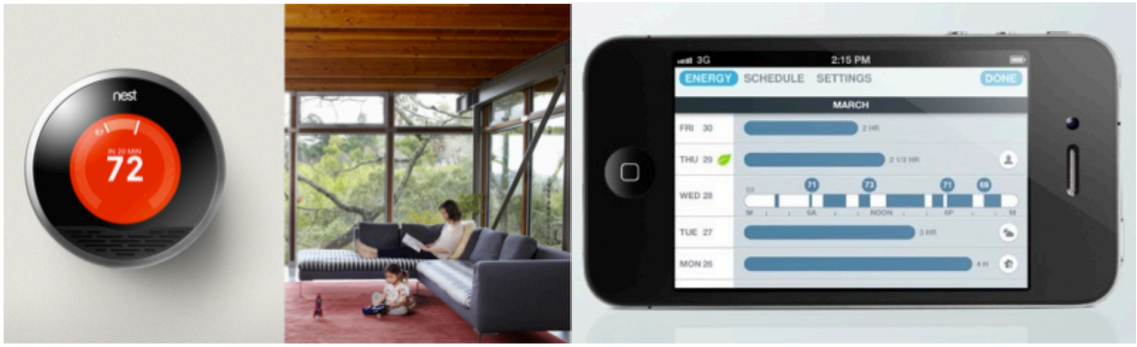
Mercedes: Learns how to drive

https://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

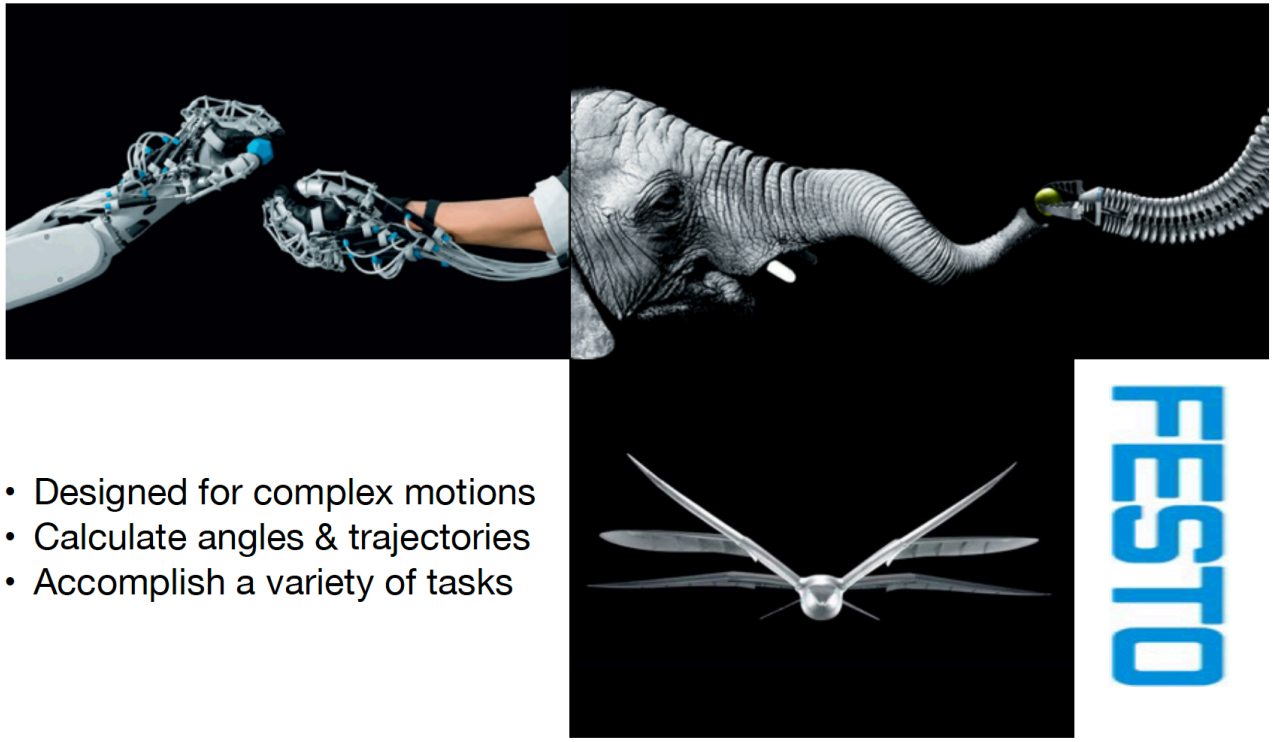
NEST: The Learning Thermostat

https://www.youtube.com/watch?v=L8TkhHgkBsg&feature=youtu.be

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

Question

- Why do we need to mimic nature?
- What are some common devices or inventions that mimic nature?

FESTO: Robots that mimic

- Vote on a video:
  - Bird - https://www.youtube.com/watch?v=nnR8fDW3Ilo
  - Kangaroo - https://www.youtube.com/watch?v=_4luJ0ZSqy8
  - Aqua Ray - https://www.youtube.com/watch?v=-vT-oidWyXE
  - Air Ray - https://www.youtube.com/watch?v=UxPzodKQays
  - Jelly Fish - https://www.youtube.com/watch?v=N-O8-N71Qcw

Geckos help clean space junk

- https://www.youtube.com/watch?v=zFdkxPF8Yul

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:

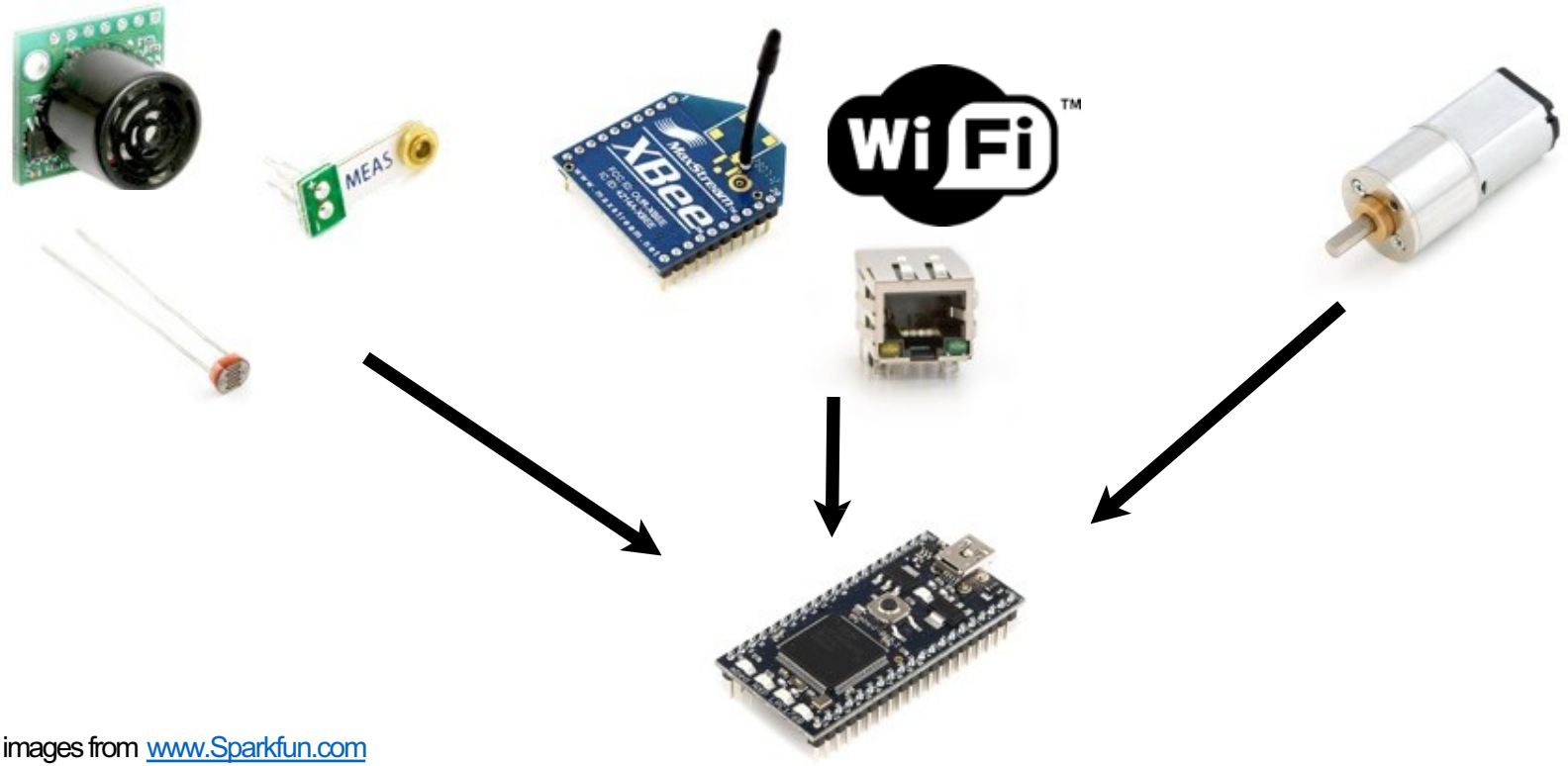
"A system of collaborating computational elements that sense or control physical entities"

Embedded Systems Can do Anything!

Sensors

Communication

Actuators



1.All images from www.Sparkfun.com

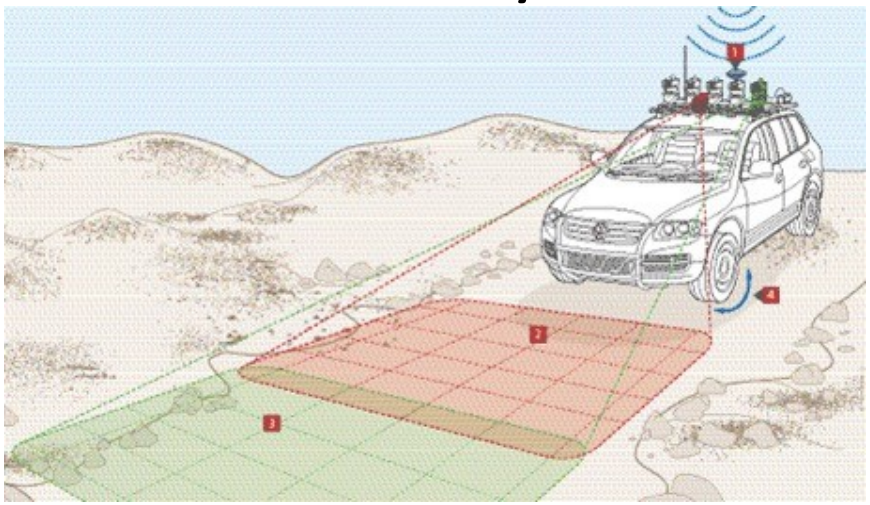
Computer / Processor

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



Scan terrain and collect data

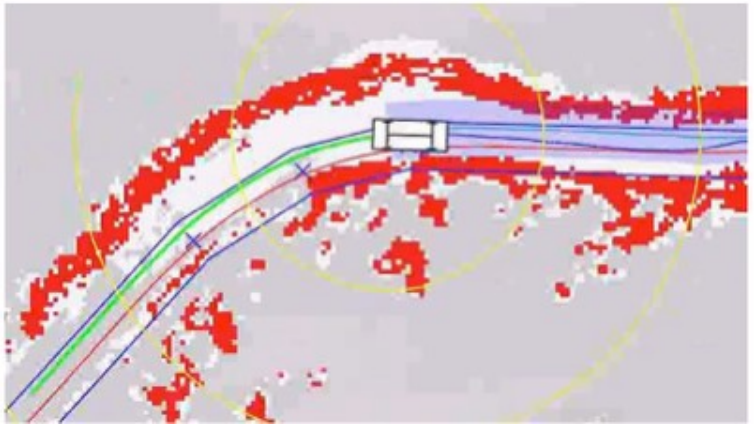
↓

Run through probabilistic models

↓

Calculate paths & obstacles

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



red: obstacles
clear: paths
gray: unknown

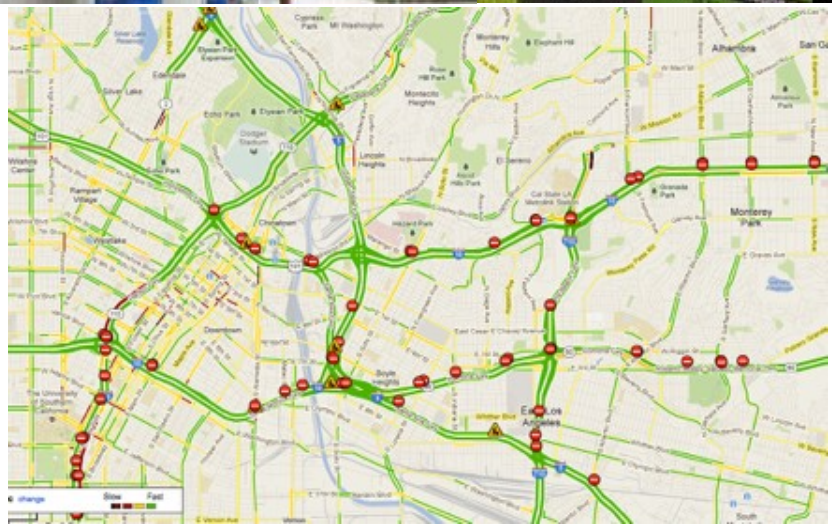
Tracking Cranes

https://www.youtube.com/watch?v=jsY5pnT0du8

cellular + shortwave radio

University of Nebraska: Cyber Physical Networking Lab

Santa Monica Parking Sensors

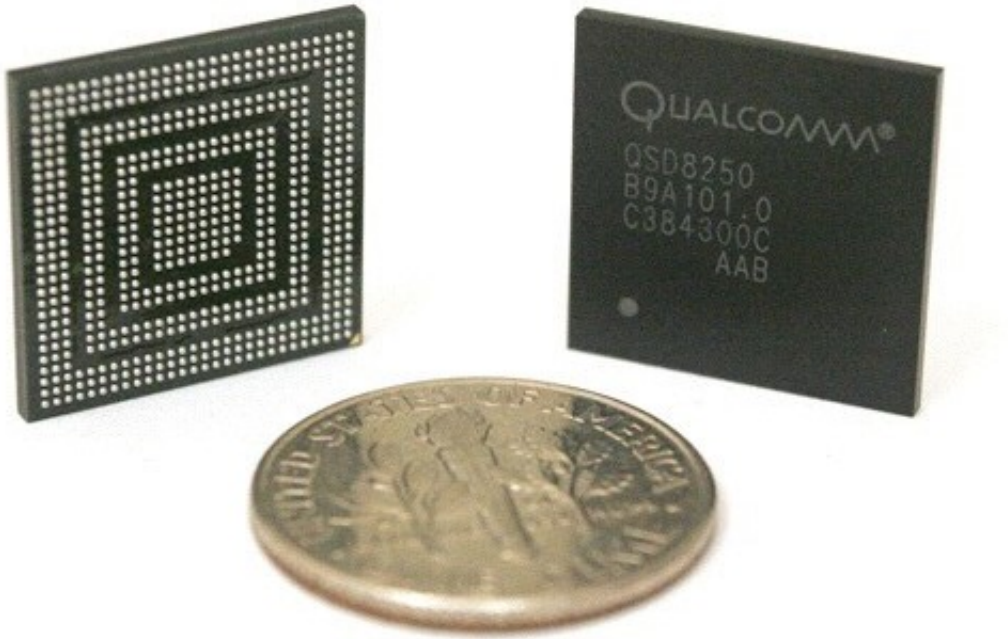


Information at your fingertips…

parking in motion
a smarter way to park

Embedded Computing

Embedded Processor



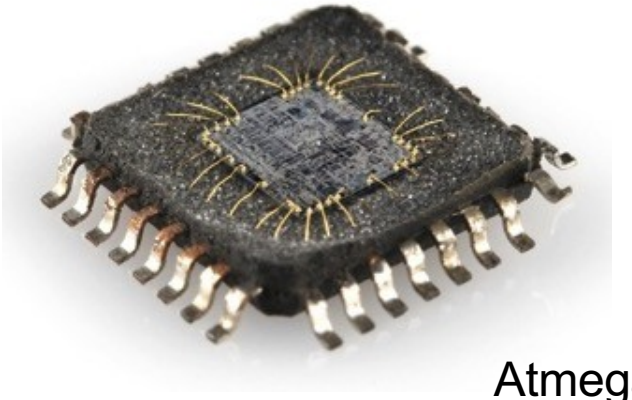
Embedded Processor: Comparison

Intel Core i7
- Speeds: 3-4 GHz
- Cores: 4+
- Memory DDR3: 32 GB
- Power: 50 ~300 Watts

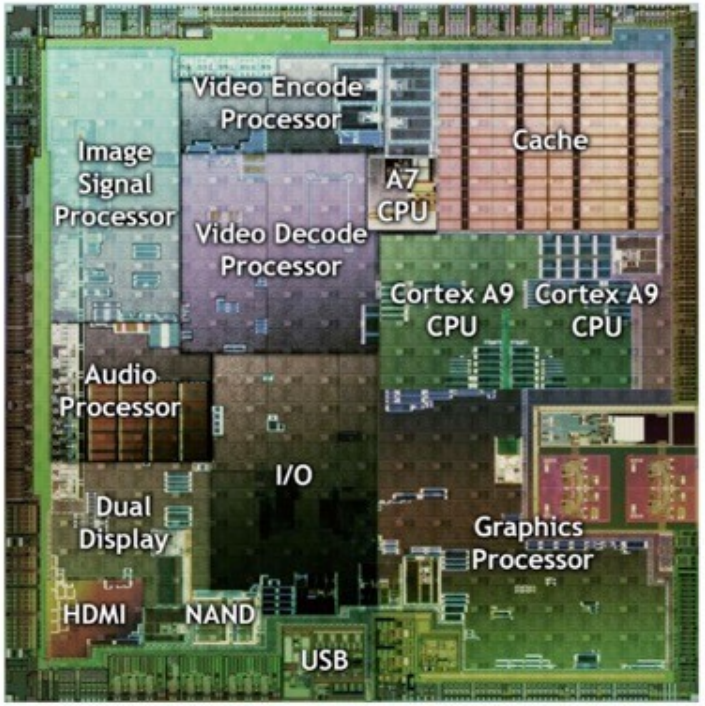
ARM Cortex-M3
- Speeds: 100 MHz
- Cores: 1
- Memory SRAM: 64 KB
- Power: ~0.5 W

Embedded Processor: Guts

NXP



Atmega

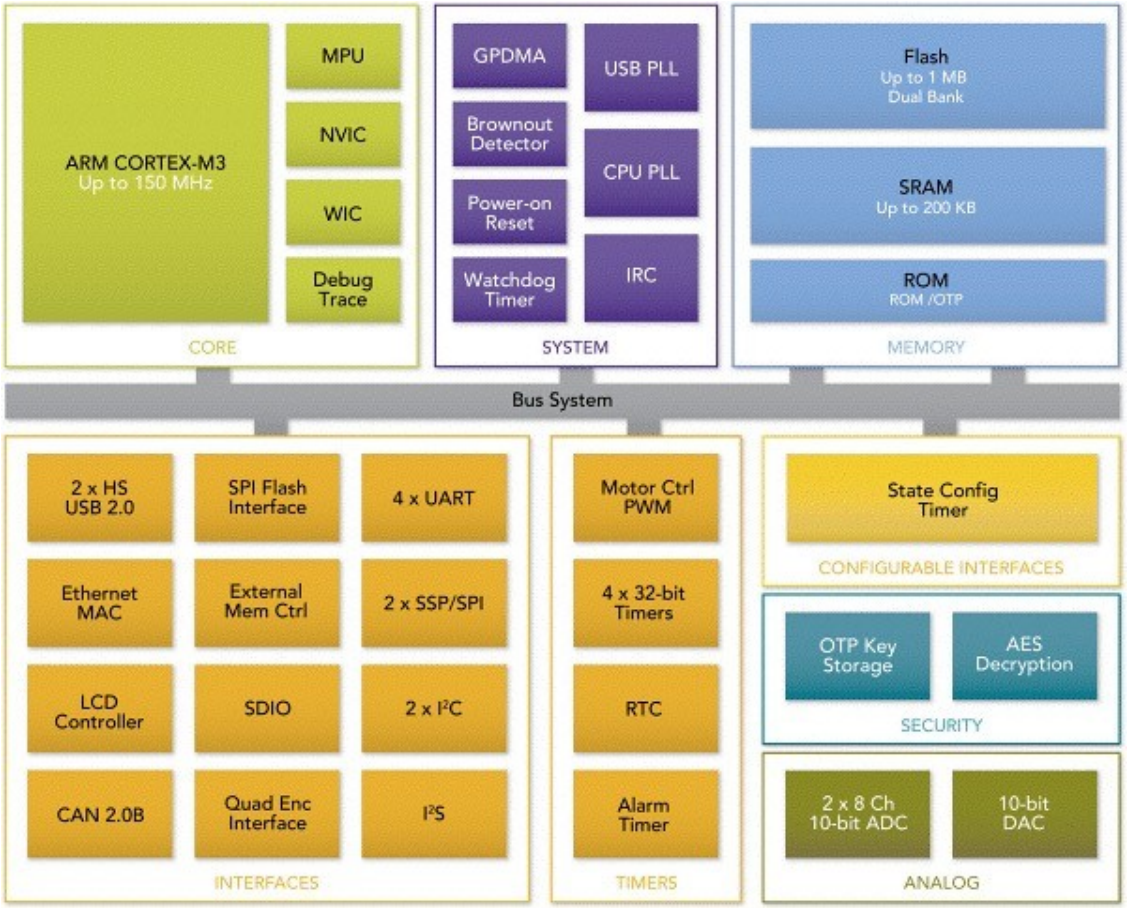


Tegra 2

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

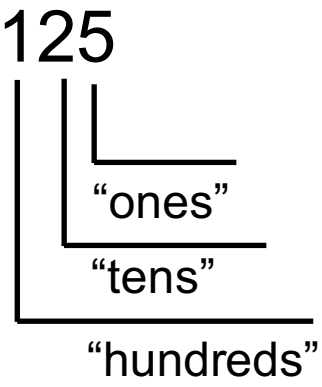
- Core computing unit
- Memory stack
- Clocks & timers
- Digital I/O
- Analog I/O
- Communication
  - USB
  - Serial
  - Ethernet
- System drivers
  - Touch screens
  - audio
  - ⋮



Number Systems

Binary Number Systems: *a primer*

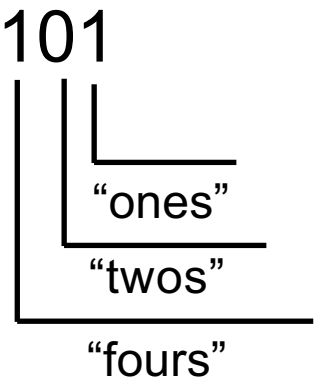
- What do we mean when we say we use a “Decimal” number system?



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

Binary Number Systems: *a primer*

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



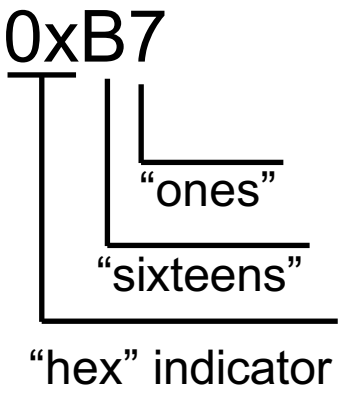
$$1(2^2) + 0(2^1) + 1(2^0) = 4 + 0 + 1$$

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)?

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

Communicating Digital Information

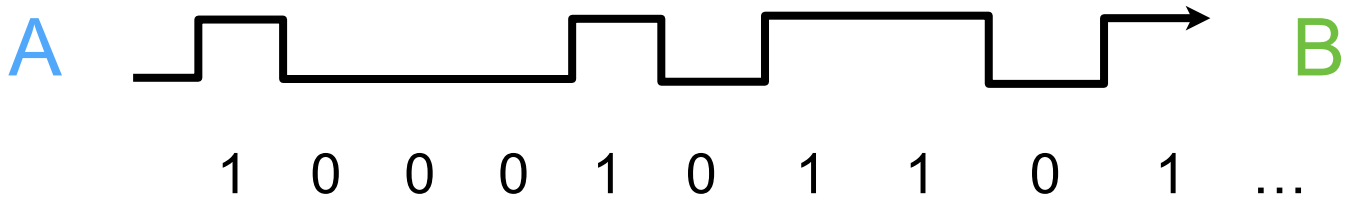
Communication

- Embedded Systems are composed of many parts, often physically separated
- How can we communicate information from one “subsystem” to another?
  - With wires? Without wires?
- 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?

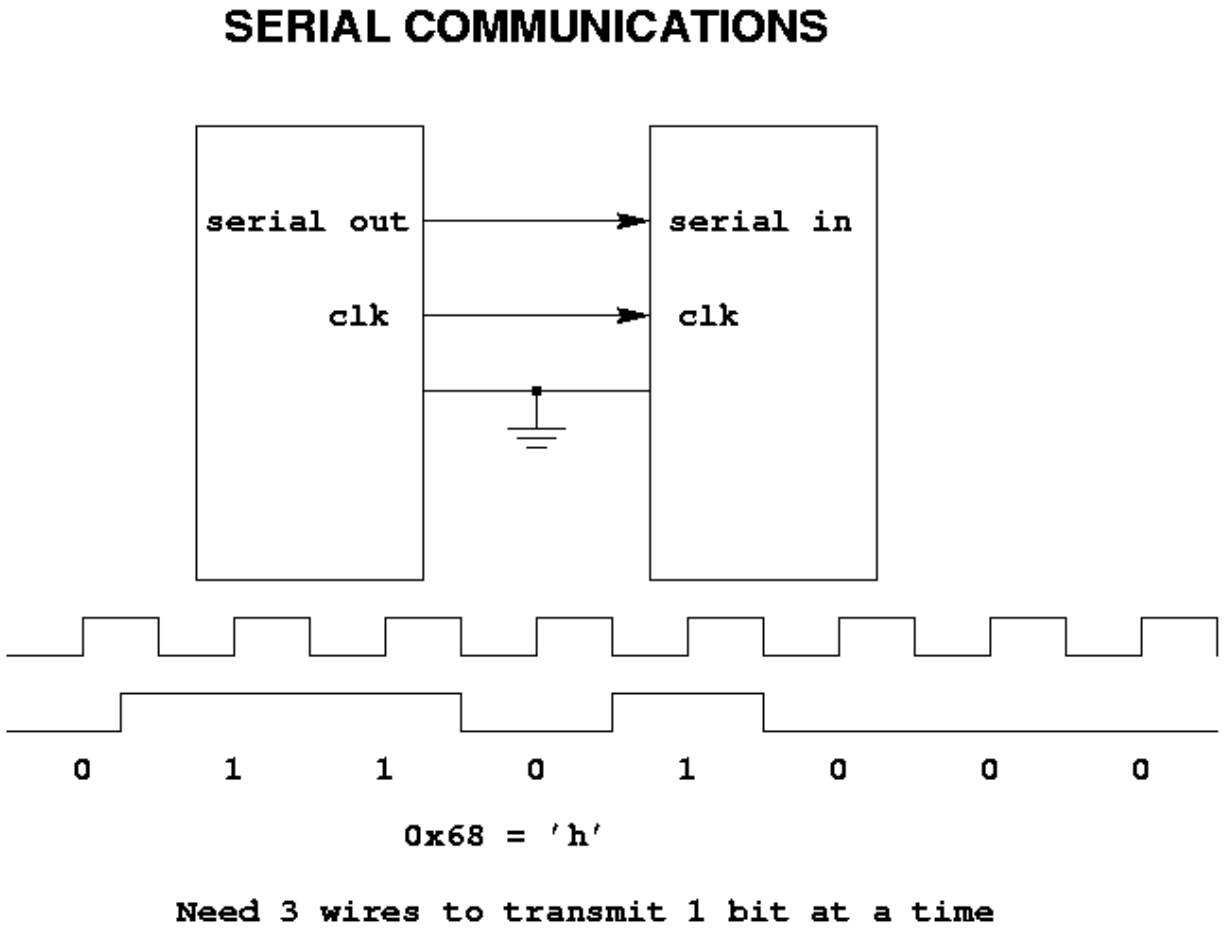
  One way:



  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.

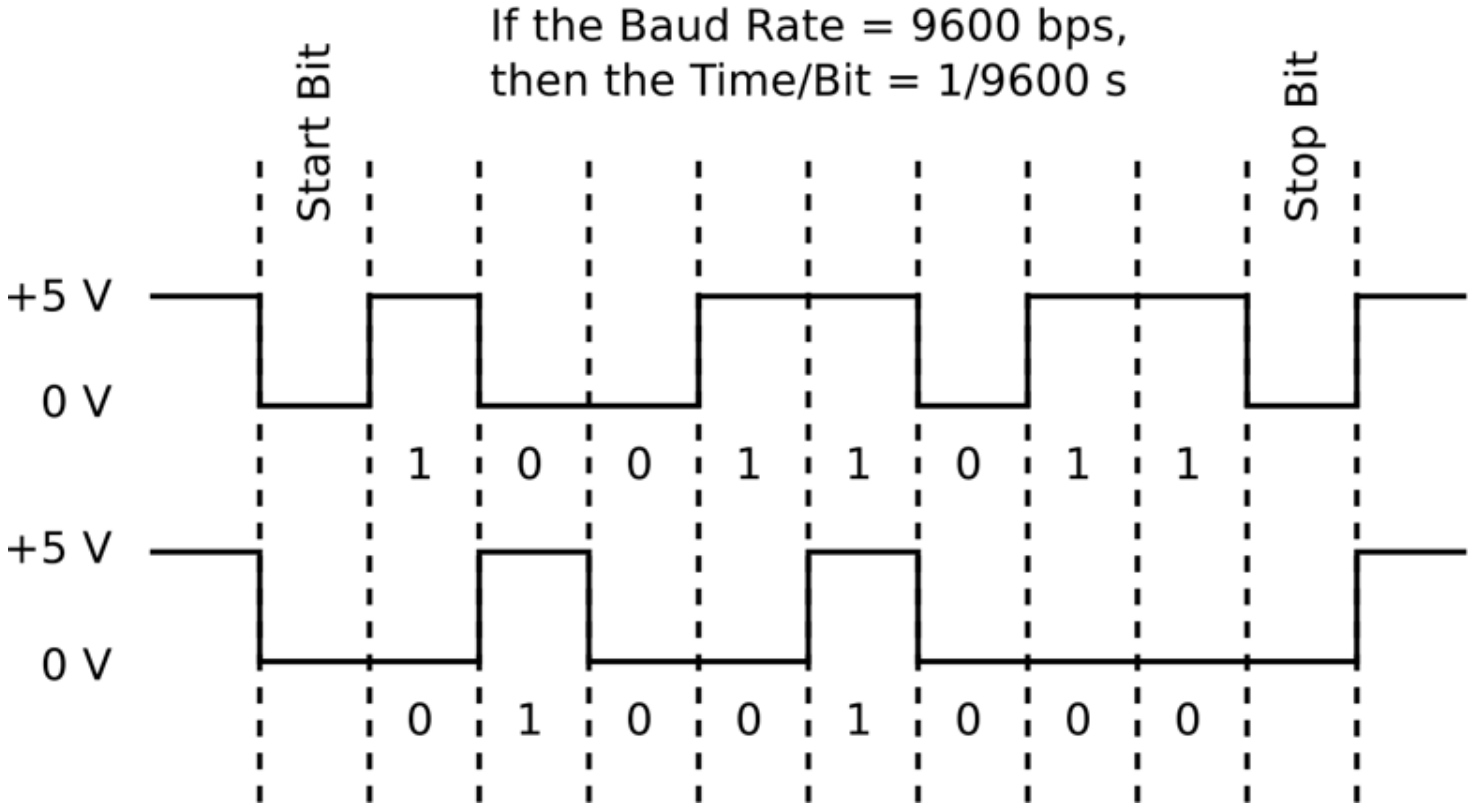
  For more information, look up UART, SPI, I^2C, & CAN protocols.

Synchronous Communication



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

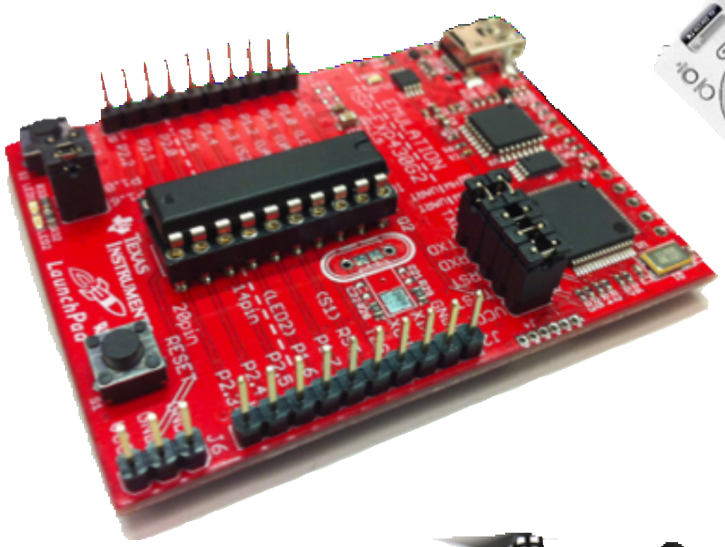
Serial “Word”

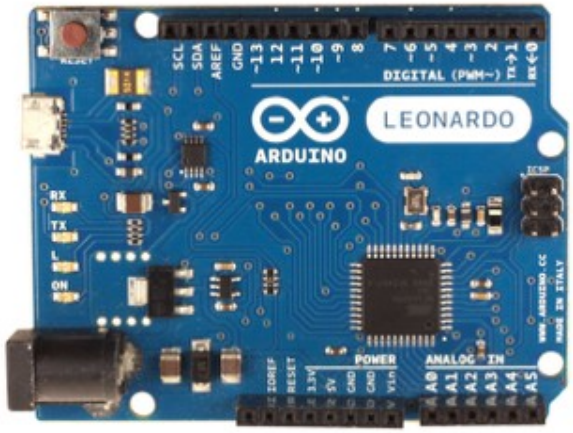

If the Baud Rate = 9600 bps,
then the Time/Bit = 1/9600 s
Start Bit
Stop Bit
+5 V
0 V
1 0 0 1 1 0 1 1
+5 V
0 V
0 1 0 0 1 0 0 0

Embedded Processor: LPC1768 Block Diagram

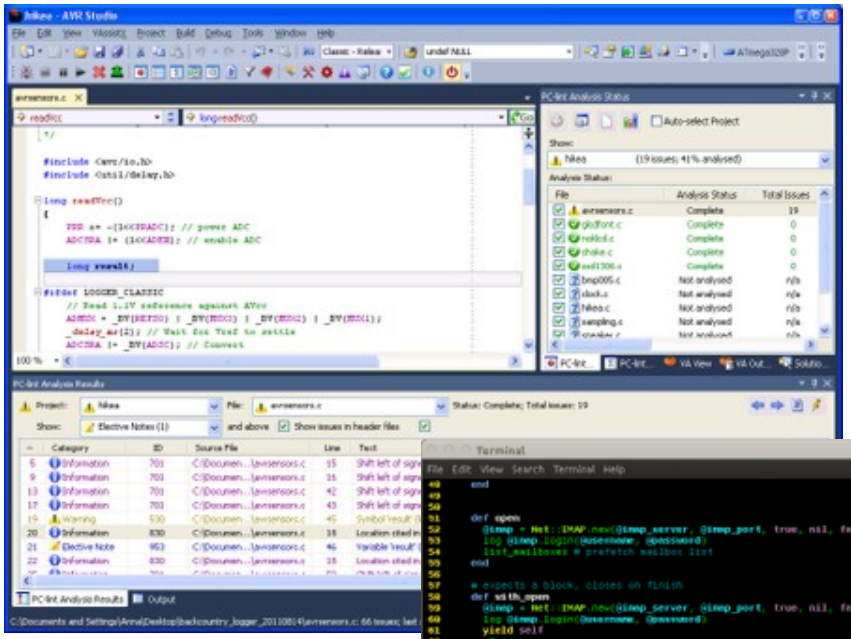
How do I develop an Embedded System?

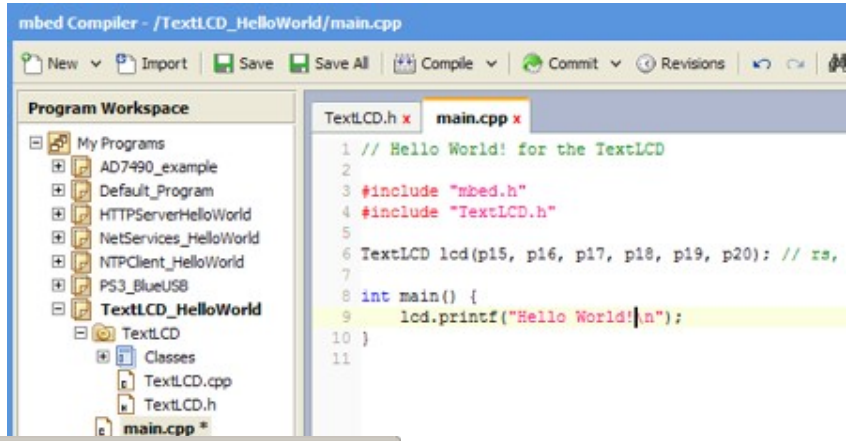
Typical *Embedded* Platforms

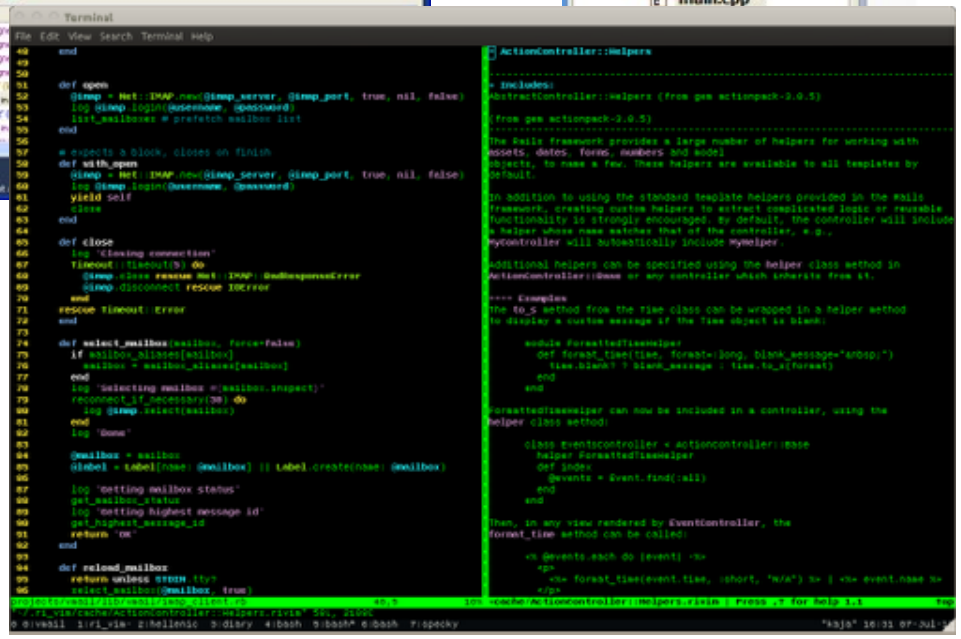




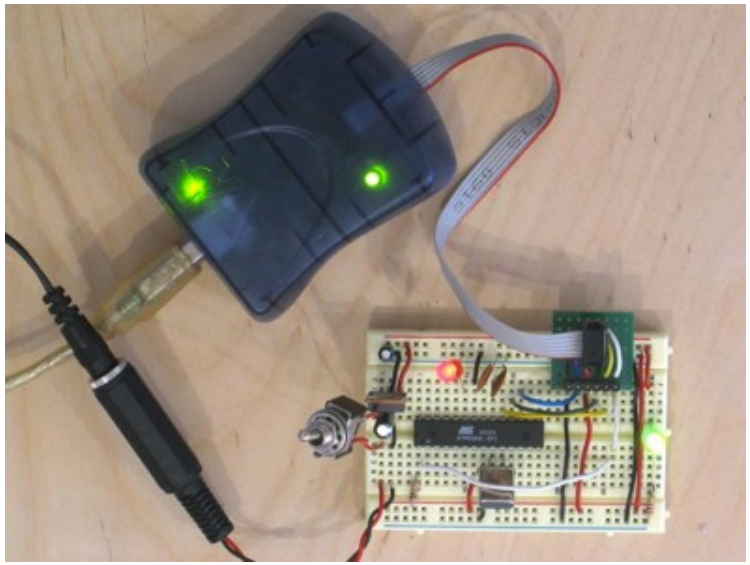
Typical Development Environments





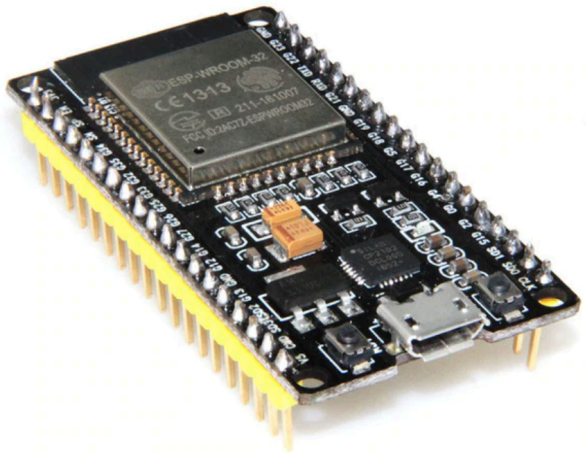


Typical Development Environments



Your Embedded Platform — “ESP32”

- Arduino IDE
- C/C++ - Arduino IDE language
- Tons of libraries
- Tons of resources online
- Wifi + Bluetooth
- On-board LED



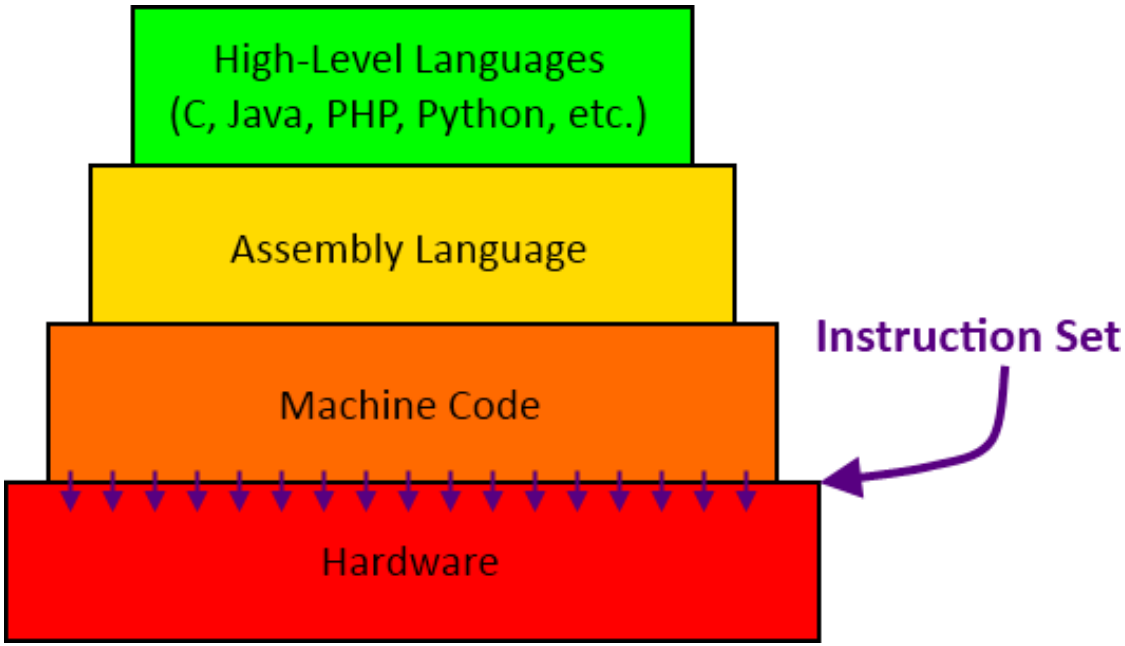
Programming an Embedded Device

- 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

LAC
C

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++ Compilation

Human Readable Language

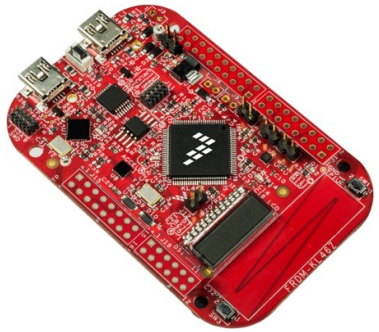
Intermediate Language

Machine Code

C++

Assembly

1001010101



C++ vs Python

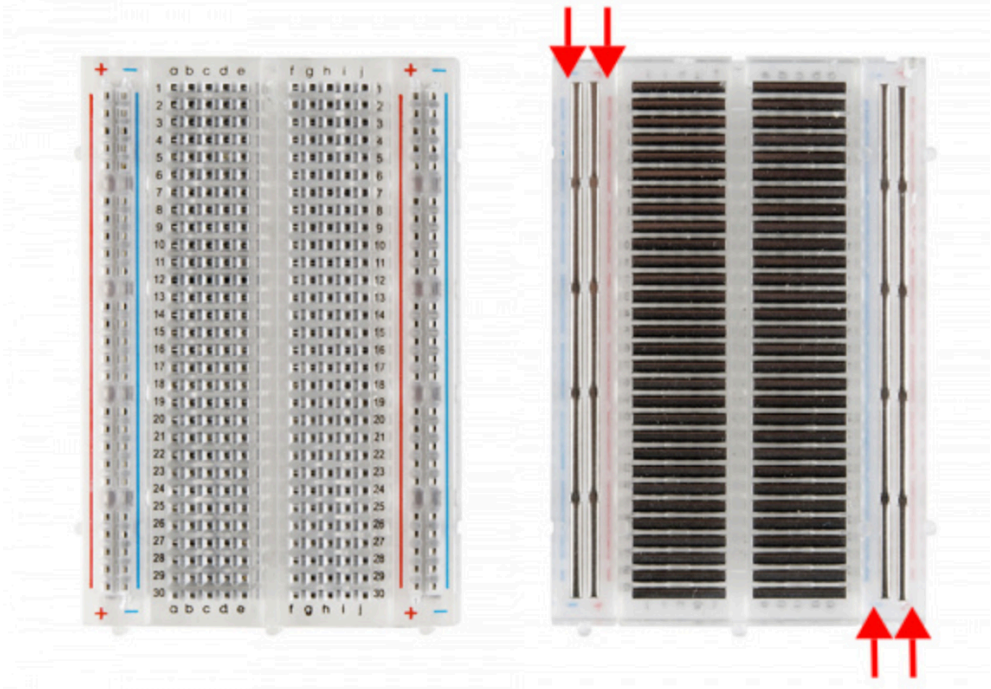
- C++ programs run faster
- Compiler optimizes code for processor
- C++ requires about 4 times more lines of code
- Python works better as a “glue” language
- Python is a good rapid prototyping language

Things you need to install

- Arduino IDE
- esp32 library
- USB drivers (if any)

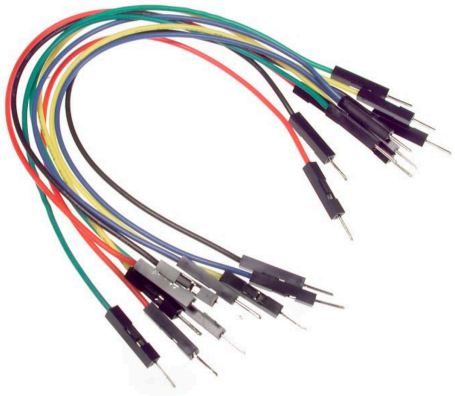
Know your components

Breadboard



Jumper cables

- Male to Male
- Male to Female
- Female to Female



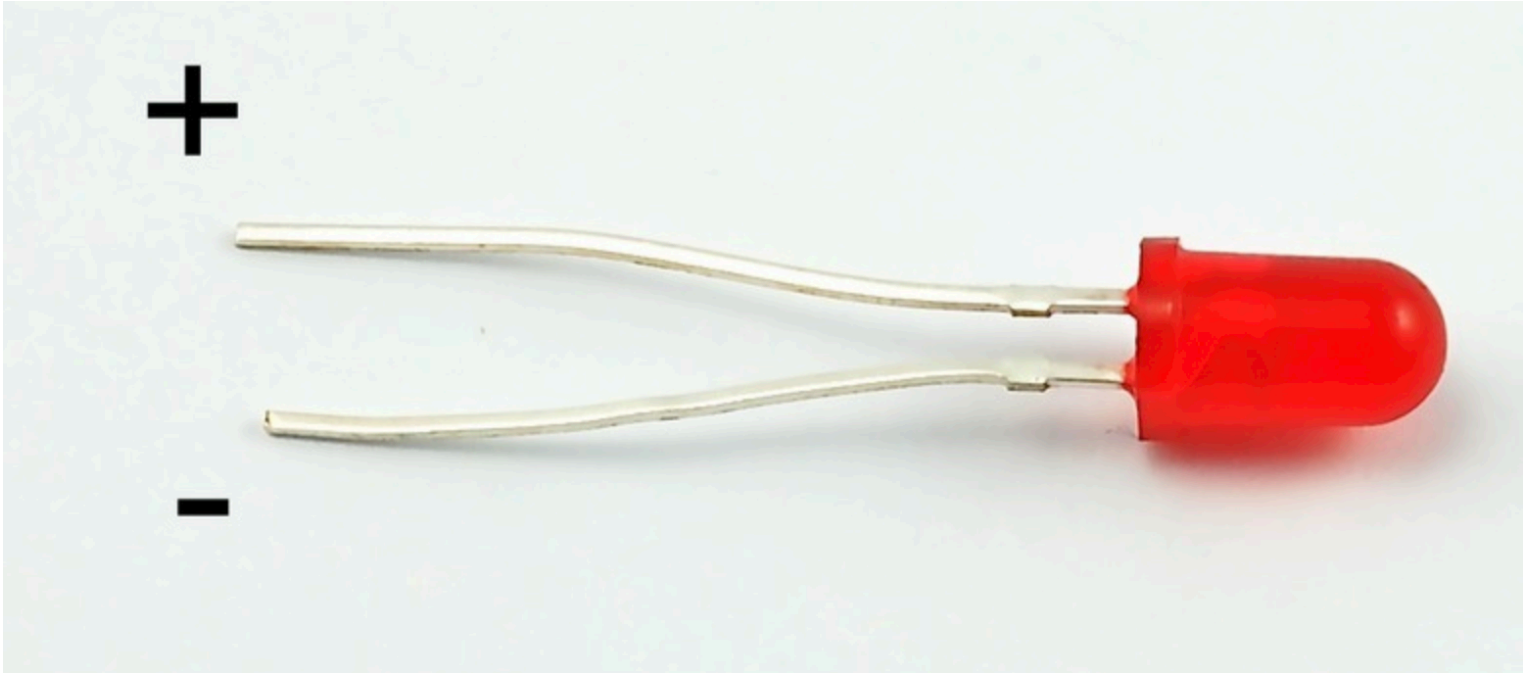
Safety

- Listen to instructions carefully before making connections.
- Double check the polarities of any connections you make.
- Don't plug in an LED without a current limiting resistor.
- Don't plug in say 9v to the 5v pin.
- Don't reverse polarity

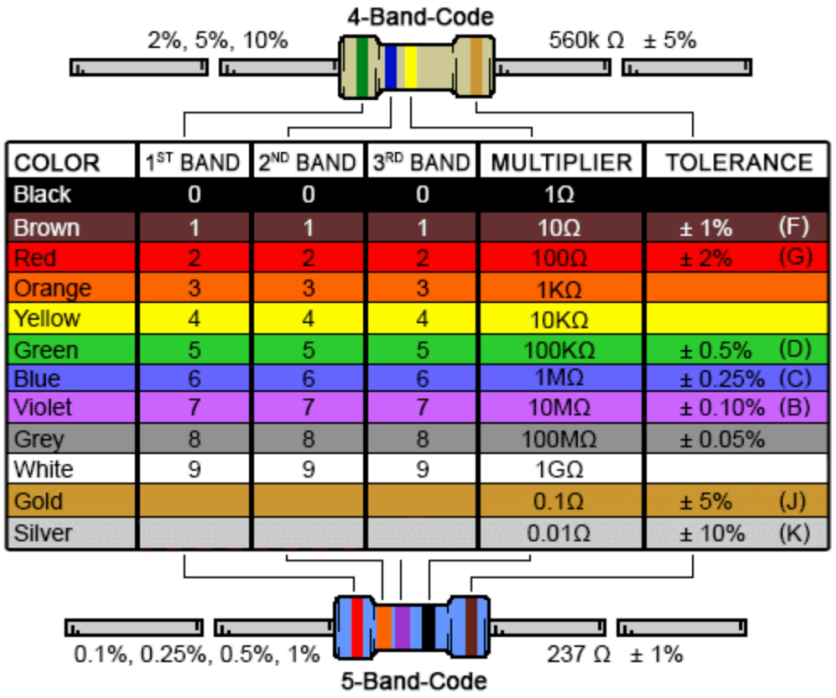
Activity-1

Blinking LED

LED

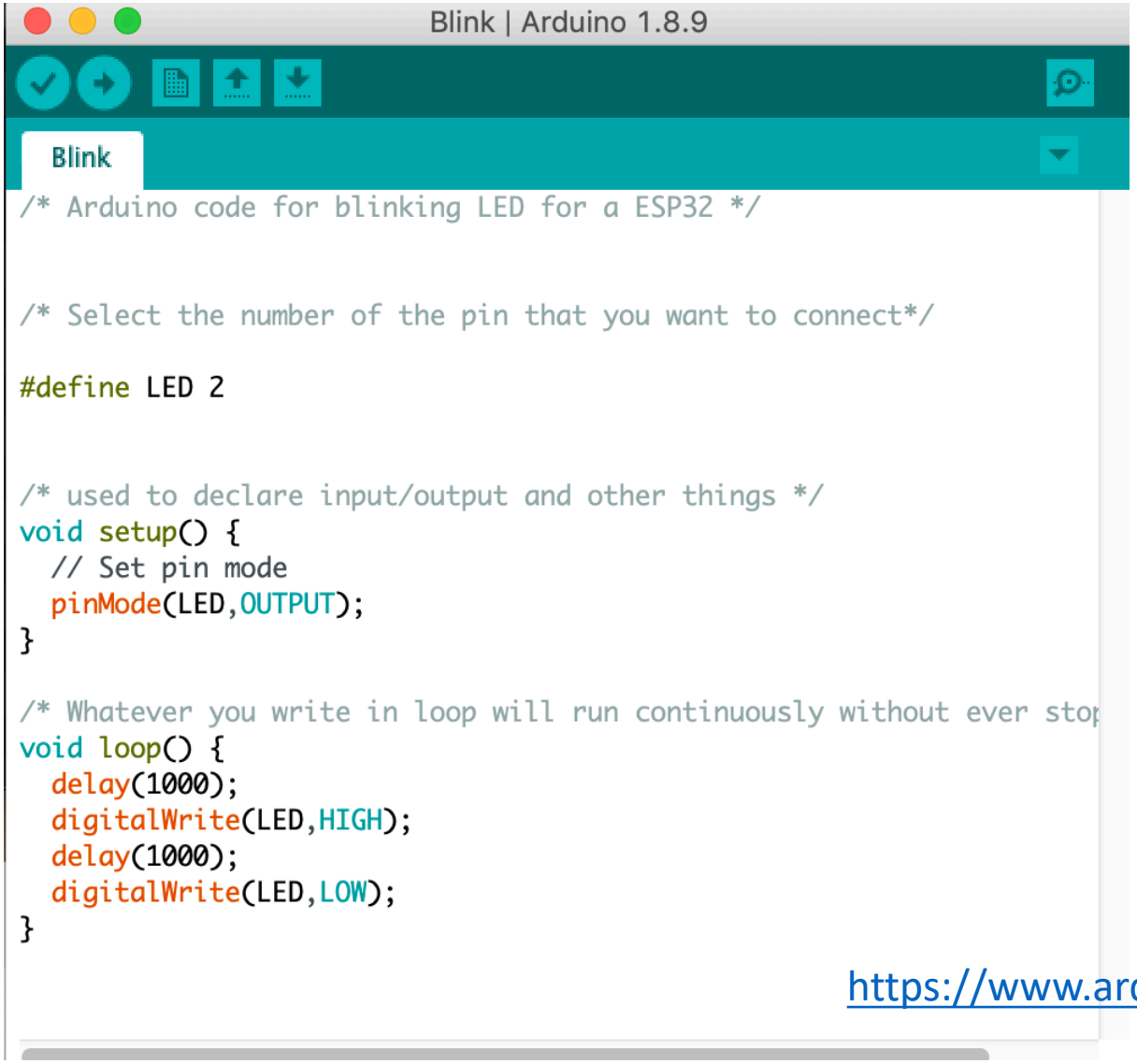


Resistor



COLOR	1ST BAND	2ND BAND	3RD BAND	MULTIPLIER	TOLERANCE
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Red	2	2	2	100Ω	± 2% (G)
Orange	3	3	3	1KΩ	
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	± 0.5% (D)
Blue	6	6	6	1MΩ	± 0.25% (C)
Violet	7	7	7	10MΩ	± 0.10% (B)
Grey	8	8	8	100MΩ	± 0.05%
White	9	9	9	1GΩ	
Gold				0.1Ω	± 5% (J)
Silver				0.01Ω	± 10% (K)

Code: “Blink”



https://www.arduino.cc/en/tutorial/blink

Connections

- Connect the resistor to Pin2 of the ESP32 board on one end and +ve terminal of the LED on other end.
- Connect the negative terminal of the LED to the ground bus.
- Wait for further instructions.

Play with the code

- Change the LED pin.
- Vary the delay time.
- Can you find the on-board LED?

Activity-2

Serial Communication

Connections

- Don’t change connections.


Serial_comm | Arduino 1.8.9
Serial_comm
/* Arduino code for blinking LED for a ESP32 */
/* Select the number of the pin that you want to connect*/
/* used to declare input/output and other things */
void setup() {
Serial.begin(9600); // Starts the serial communication
}
/* Whatever you write in loop will run continuously without ever stop
void loop() {
Serial.println("Hi! I am printing stuff on the serial monitor");
delay(3000);
Serial.println("Oops! I erased it");
delay(3000);
}

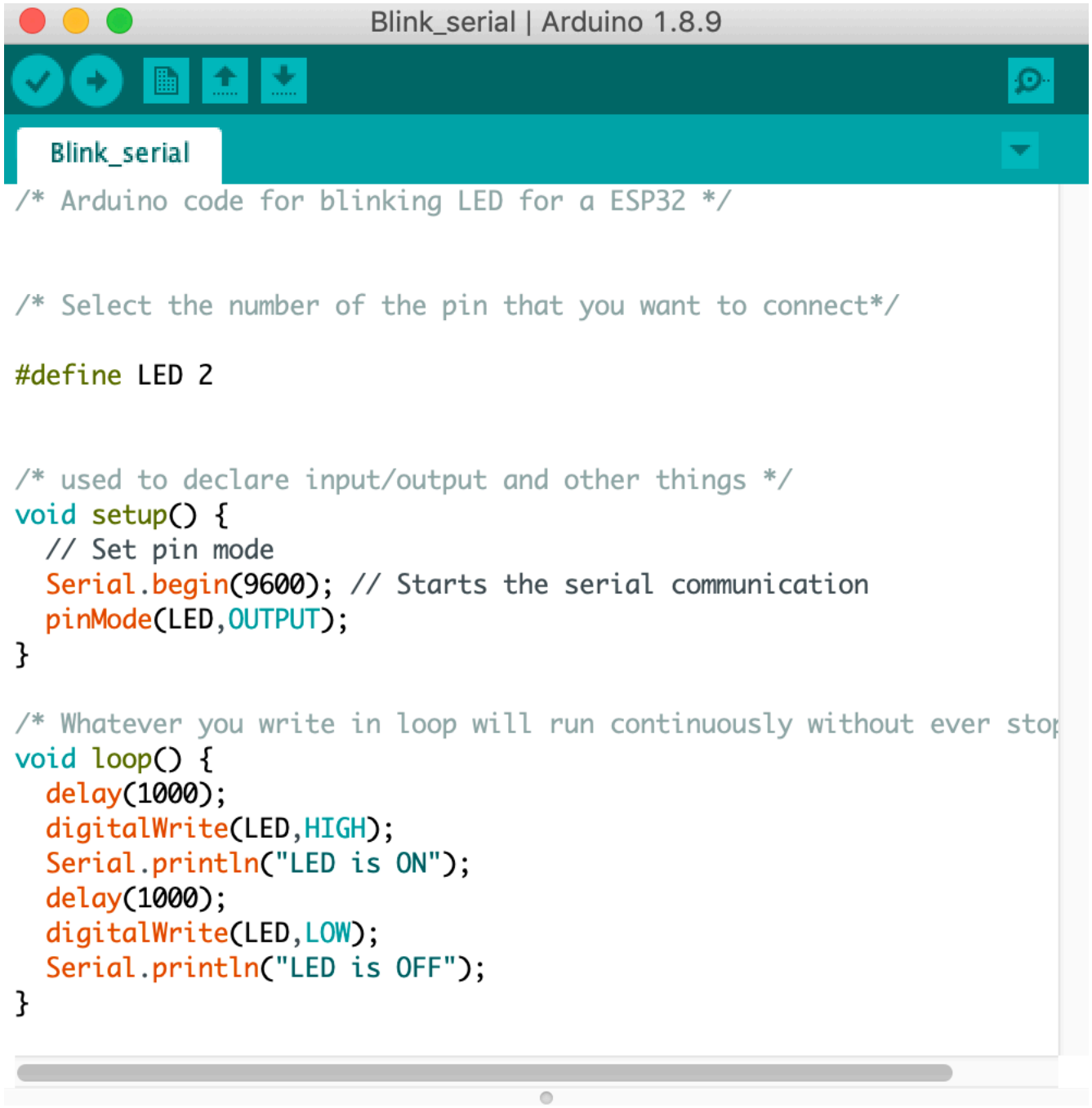
Activity-3

LED Blink with Serial Communication

What to do?

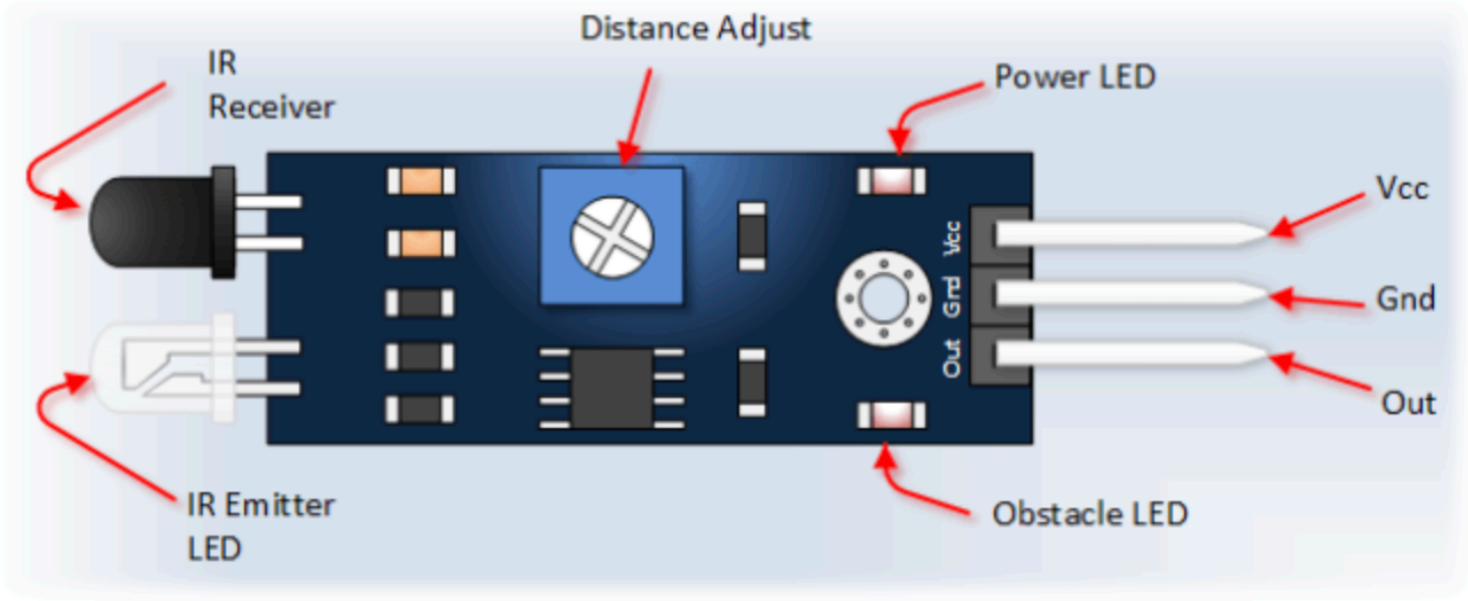
- You know how to blink an LED.
- You know how to do serial communication.
- Can you combine the two?
  - When the LED is ON -> Print “LED is ON” on the serial monitor
  - When the LED is OFF -> Print “LED is OFF” on the serial monitor

Let’s go over the solution code


Blink_serial | Arduino 1.8.9
Blink_serial
/* Arduino code for blinking LED for a ESP32 */
/* Select the number of the pin that you want to connect*/
#define LED 2
/* used to declare input/output and other things */
void setup() {
// Set pin mode
Serial.begin(9600); // Starts the serial communication
pinMode(LED,OUTPUT);
}
/* Whatever you write in loop will run continuously without ever stop
void loop() {
delay(1000);
digitalWrite(LED,HIGH);
Serial.println("LED is ON");
delay(1000);
digitalWrite(LED,LOW);
Serial.println("LED is OFF");
}

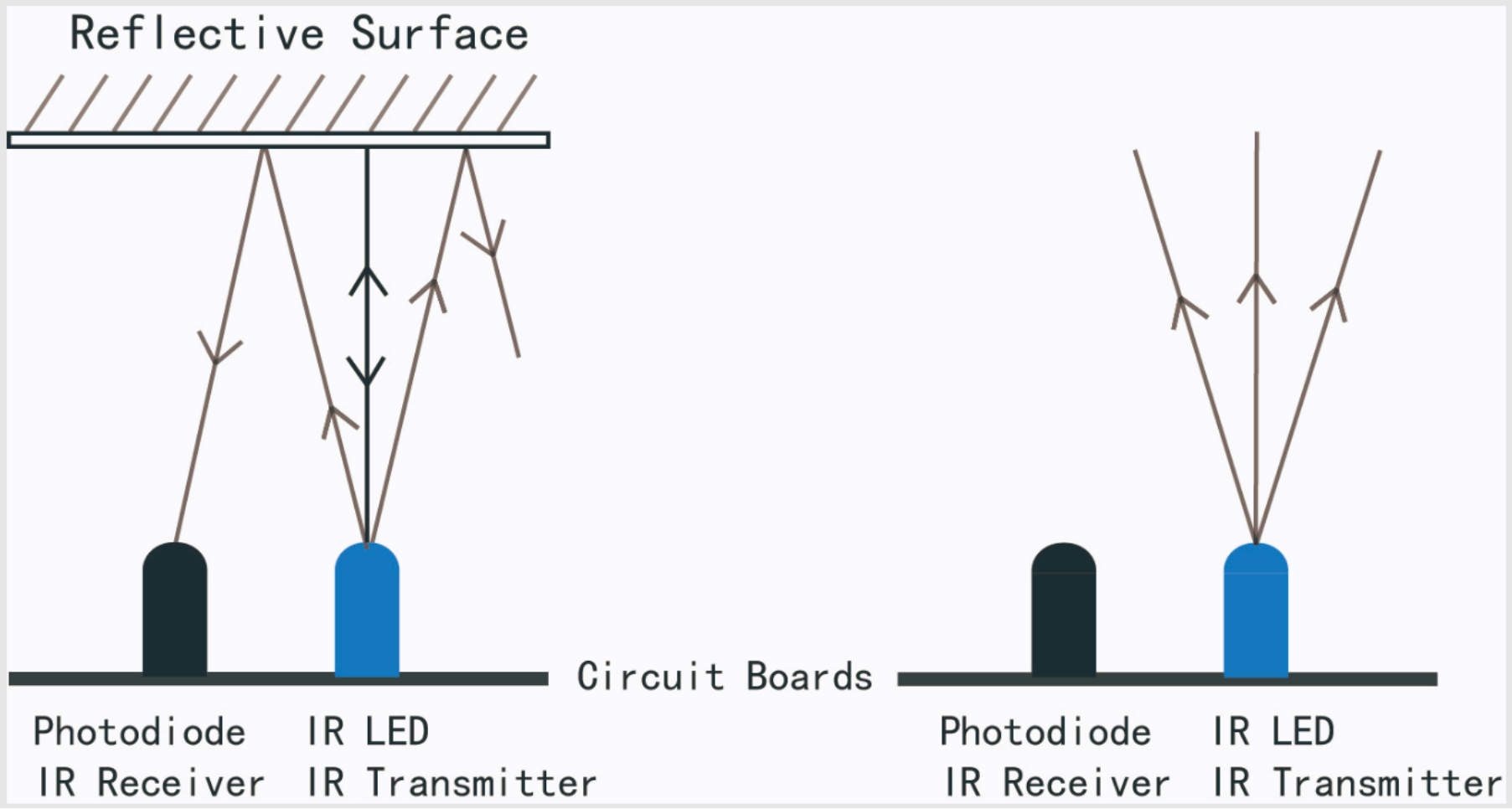
Activity-4

Obstacle detection using an IR sensor



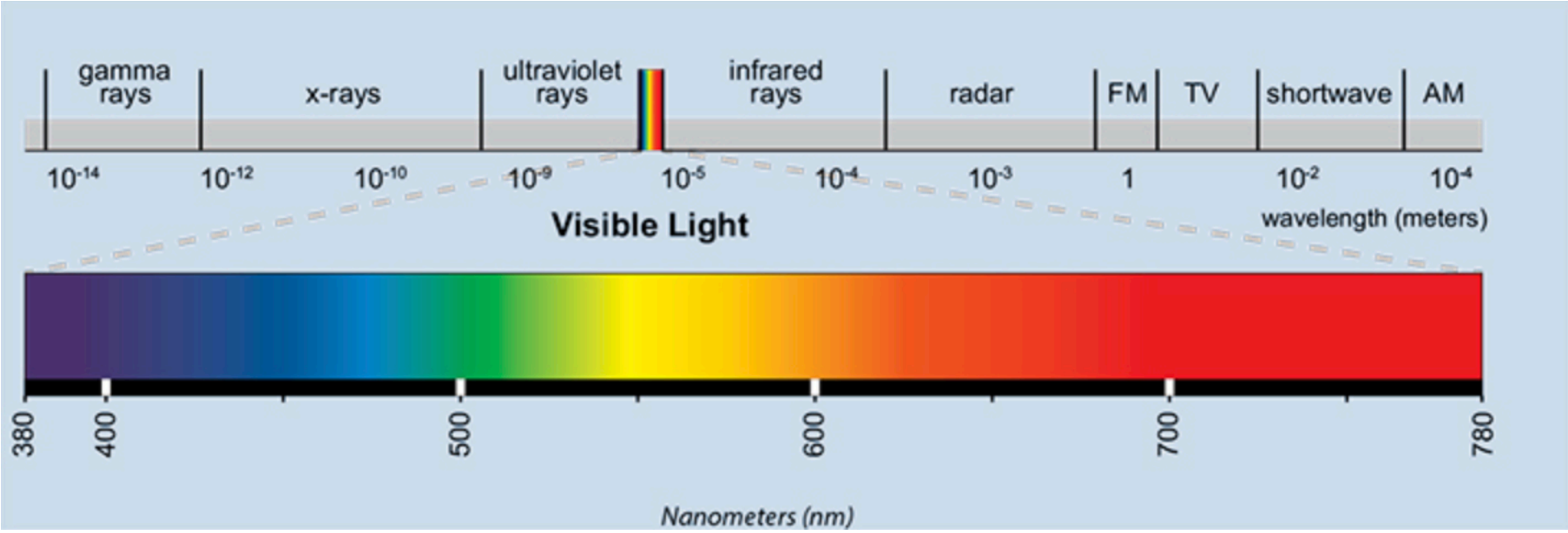
Pin, Control Indicator	Description
Vcc	3.3 to 5 Vdc Supply Input
Gnd	Ground Input
Out	Output that goes low when obstacle is in range
Power LED	Illuminates when power is applied
Obstacle LED	Illuminates when obstacle is detected
Distance Adjust	Adjust detection distance. CCW decreases distance. CW increases distance.
IR Emitter	Infrared emitter LED
IR Receiver	Infrared receiver that receives signal transmitted by Infrared emitter.

Working Principle



http://osoyoo.com/

What is IR?


gamma rays
x-rays
ultraviolet rays
infrared rays
radar
FM
TV
shortwave
AM
10-14
10-12
10-10
10-9
10-5
10-4
10-3
1
10-2
10-4
wavelength (meters)
Visible Light
380
400
500
600
700
780
Nanometers (nm)

Code:


IR2 | Arduino 1.8.9
IR2
const int ProxSensor=2; //Declaring where the Out pin from the sensor
void setup() {
// You need to define something here
pinMode(ProxSensor,INPUT); // then we have the out pin from the mod
}
void loop() {
if(digitalRead(ProxSensor) == HIGH) //Check the sensor output
{
//Print "All clear" on the serial monitor
}
else
{
// Print "Obstacle Detected" on the serial monitor
}
delay(100);
}

Play with the potentiometer

- Try and turn it around to see a change in distance.

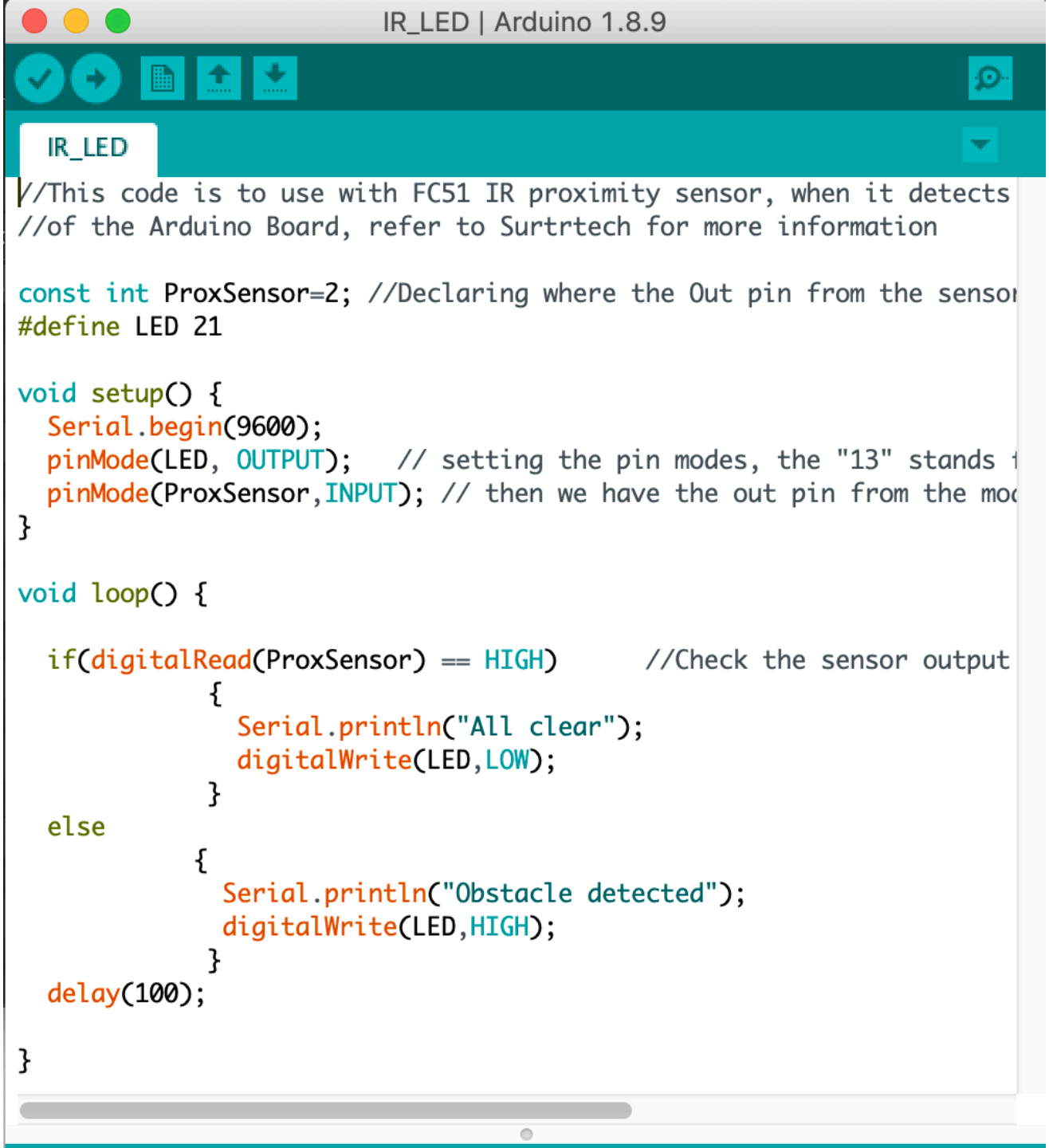
Activity-5

Obstacle detection using an IR sensor and LED Blink

What to do?

- You know how to blink an LED.
- You know how to do serial communication.
- You know how does an IR sensor work.
- Combine all 3. The LED should blink when the sensor detects an obstacle.

Let’s go over the solution code


IR_LED | Arduino 1.8.9
IR_LED
//This code is to use with FC51 IR proximity sensor, when it detects
//of the Arduino Board, refer to Surtrtech for more information
const int ProxSensor=2; //Declaring where the Out pin from the sensor
#define LED 21
void setup() {
Serial.begin(9600);
pinMode(LED, OUTPUT); // setting the pin modes, the "13" stands
pinMode(ProxSensor,INPUT); // then we have the out pin from the mod
}
void loop() {
if(digitalRead(ProxSensor) == HIGH) //Check the sensor output
{
Serial.println("All clear");
digitalWrite(LED,LOW);
}
else
{
Serial.println("Obstacle detected");
digitalWrite(LED,HIGH);
}
delay(100);
}

Internet of Things (IoT)

What is IoT?

- Forbes: *Simply put, this is the concept of basically connecting any device with an on and off switch to the Internet (and/or to each other). This includes everything from cellphones, coffee makers, washing machines, headphones, lamps, wearable devices and almost anything else you can think of.*

https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#1584ce6c1d09

Embedded, Wearable Devices

Mobile Devices

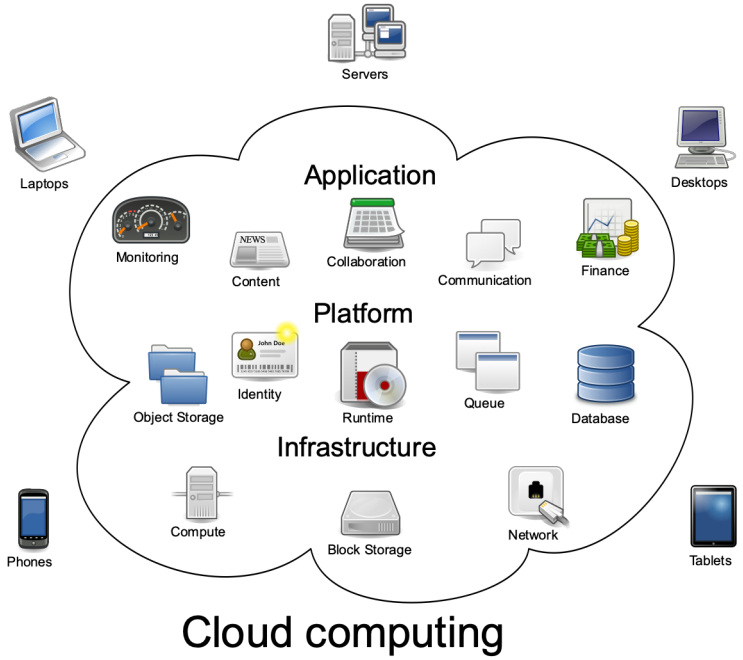
Planetary-scale Network

Cloud Services

Taken from Prof. Mani Srivastava's lecture notes (ECE M202A at UCLA)

Cloud Computing

- Wikipedia: *Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user*



Life simplified with IoT

- https://www.youtube.com/watch?v=NjYTzvAVozo
- https://www.youtube.com/watch?v=QSIPNhOiMoE

Self Driving

- https://www.youtube.com/watch?v=jWreyC2l-dw
- Cars: https://www.youtube.com/watch?v=ugNJJf2QW0E
- Trucks: https://www.youtube.com/watch?v=HBU712__R_c

Activity-6

Ultrasonic sensor for distance measurement