

CT106-3-M-BIA - Building IoT Applications CT127-3-M-ODL - BIA - Building IoT Applications

Topic 1 - Introduction to the Internet of Things (IoT)

TOPIC LEARNING OUTCOMES



At the end of this topic, you should be able to:

- Define IoT and explain its concept, emphasizing the network of interconnected physical objects and their ability to collect and exchange data over the Internet.
- Understand the origins and evolution of IoT, including its early beginnings and the factors that contributed to its growth and widespread adoption.
- Identify and describe the key components of IoT hardware, such as sensors, actuators, microcontrollers and microprocessors.
- Explore prototyping tools used in IoT development that facilitate the creation of IoT prototypes and proofs-of-concept.
- Explore data acquisition, analysis and visualization tools used to collect, analyze, represent and interpret IoT data, including graphical interfaces and dashboards.



Contents & Structure

- What is IoT?
- How did it start?
- IoT Hardware:
 - Sensors and Actuators
 - Microcontrollers and Microprocessors
- IoT Software:
 - Prototyping Tools
 - Professional Programming IDEs
 - Data Acquisition and Analysis Software
 - Visualisation Tools

What is Internet of Things?



The Internet of Things (IoT) refers to a network of interconnected physical devices, vehicles, appliances, and other objects embedded with sensors, software, and network connectivity that enables them to collect and exchange data.



Source: https://inventrom.wordpress.com/2014/11/27/the-thing-in-interne t-of-things/

What is the Internet of Things (IoT)





Image from http://www.cchc.cl/informacion-a-la-comunidad/industria-de-la-construccion/personaje/

SLIDE 5

5

Sensors

Measure values

Send raw data

Low power





Local Processing and Local Storage

Get data from sensors

Process

Send some data to

Edge/Fog Computing



7



Network and Internet

IoT Gateway

Gathers data from sensors

Gateway Protocols

- 6LoPAN
- LoRaWAN
- BLE

Internet Protocols

- CoAP
- MQTT
- HTTP
- XMPP



8

Cloud Processing and Storage

Aggregate Data

Storage

Inferences





IoT Network



How did it start



open source hardware

CTI06-3-M-BIA - Building IoT Applications

Introduction to the Internet of Things

11 SLIDE 11

Origins of the Internet of Things (IoT)



- Embedded Systems: The concept of embedding computing capabilities into physical objects can be traced back to the 1970s with the emergence of embedded systems. These systems were designed to control and monitor specific functions in various devices and machines.
- Machine-to-Machine (M2M) Communication: In the 1980s, the idea of machines communicating with each other without human intervention gained traction. M2M communication aimed to enable devices to share data and work collaboratively in industrial settings.
- Internet of Things (IoT): The term "Internet of Things" was coined in the late 1990s but the evolution of IoT gained momentum with advancements in technology. The proliferation of wireless networks, the miniaturization of sensors, and the development of low-power microcontrollers played pivotal roles.

Concepts close to IoT



Ambient Intelligence (AmI):

- AmI leverages IoT technologies, sensors, and artificial intelligence to create smart environments that anticipate and respond to human behaviour.
- It refers to a vision where technology seamlessly integrates into our surroundings, creating intelligent and responsive environments.

Ubiquitous Computing

- Ubiquitous Computing (or pervasive computing) involves the idea of having numerous interconnected devices and sensors embedded in our environment, allowing for ubiquitous access to computing resources and services.
- It emphasizes the integration of computing technology into everyday objects and surroundings to enable seamless interactions and information access.

Why is this happening now?



Affordable hardware

• Costs for actuators & sensors have been cut into half over the last ten years.

Ubiquitous & cheap mobility

• Cost for mobile devices, bandwidth and data processing have declined as much as 97% over the last ten years.

Availability of supporting tools

• Big data tools & cloud-based infrastructure have become widely available and fairly sophisticated.

Why is this happening now?



Smaller, but more powerful hardware

• Form factors of hardware (sensors, communication technology, etc.) have shrunk to millimeter or even nanometer levels. Now you can get a low-energy, state-of-the-art sensor with ubiquitous connectivity the size of your fingertip.

Mass market awareness

• The vision of a connected world has reached such a followership that companies have initiated IoT development & marketing budgets.

IoT Hardware



• U

Microcontroller

Small programmable device

Easy connectable





Arduino

Small programmable device Easy

connectable

Is open source

Has a simple to use software

Only around 4 simultaneous networking connections



Microprocessor

High processing power and memory

Higher power consumption

Expensive





Raspberry Pi

Computer

Runs Linux OS called Rasbean

More software-oriented programming

Full Networking System



Good for sensors





Arduino ATmega328







LaunchPad MSP430

21 SLIDE 21

Good for some sensors and processing





STM32 ARM Cortex M0, M3, M4

Particle ARM WiFi Internet



Espruino ARM Javascript

Good for processing and network





Raspberry Pi 900 MHz ARM, GPU 1 GB RAM Compute Module Intel® Galileo 400 MHz Quark x86 256 MB RAM





Intel® Edison 1 GHz Dual Core Atom x86 1 GB RAM WiFi BLE 4 GB Flash

Good for processing and network





UDOO Neo i.MX 6 Solo ARM, GPU ARM M4 512 MB or 1 GB RAM



Beaglebone Black 1 GHz ARM, GPU 512 MB RAM 4 GB Flash



Parallella 1 GHz Dual Core Zynq ARM 16 or 64 Epiphany CPUs

loT Software

A · P · U Aia Pacific UNIVERSITY of Technology a Innovation

Prototyping Tools

Programming Languages

Data Storage and Analysis

Solutions Builders

Prototyping

ARDUINO

00	Blink Arduino 1.0.3	
		P
Blink		
7* Blink Turns on an LED on for one	e second, then off for one second, repeatedly.	
This example code is in th */	e public domain.	
<pre>// Pin 13 has an LED connect // give it a name: int led = 13;</pre>	ed on most Arduino boards.	
<pre>// the setup routine runs or void setup() { // initialize the digital pinMode(led, OUTPUT); }</pre>	ice when you press reset: pin as an output.	
<pre>// the loop routine runs ove void loop()[(digitalWrite(led, HIGH); delay(1000); digitalWrite(led, LOW); delay(1000); }</pre>	r and over again forever: // turn the LED on (HIGH is the voltage level) // wait for a second // turn the LED off by making the voltage LOW // wait for a second	
	*	
24	Arduino Mega (ATmega1280) on /dev/tty.usbserial-A6	00enbz

WYLIODRIN STUDIO



Professional Programming IDEs

ECLIPSE

😣 🖻 💿 C/C++ - nanomind/src/main.c	- Eclipse					
File Edit Source Refactor Navigate	Search Run Project Win	ndow Help				
C* 🖩 🗟 🚔 🗋 💣 🚳 🛛 🐨	· · · · · · · · · · · · · · · · · · ·	• 💁 🖉 😂 😂 🖋 🛛) > • • • •-	• §∥• *⊳ ⇔ • ⇔∗		🖻 🗟 C/C++
Project Explorer 🕱 🗧 🗖	Dower.c Dower.c	fpga.c spi_wrapper.c	🗎 install.txt	le *main.c ⊠	- 8	🗄 Outli 🛿 💿 Make 🍰 Call 📄
<pre></pre>	/** NanoMind3 @uthor Johan De Cl Copyright 2011 Gent USING SRAM MEMORY: To store a variable static_atribute To place a function 	<pre>laville Christiansen Space ApS. All rights reser e in internal SRAM use:((section(*.sram.data*))) n in internal SRAM use: tion(*.sram.text*))) p.m.hb e.h supervisor.hb ccs/csp_if_can.hb sp_if_l2c.hb cs_if_lic.hb cs_if_</pre>	ved. rror Log =ö Progre Writable	ss) 🛷 Search 🚵 Git Stagi Smart Insert 15: 2	ng = Resc	If a b to be a set of the set
1						

VIM

			VIM					
~/Projects/exaile				* Sorted	by mru			
. py	def get	collection_by_loc(loc):					In the second second second	
NG 1e		gets the collection by a location.		20 #	collection.py	/home/synic	/Projects/exaile/working	
agma keep				18	queue.py	/home/synic	/Projects/exaile/working	/xl/player
					pipe.py	/home/synic	/Projects/exaile/vorking	/xl/player
		return: collection at location or No	one About Exaile	12	engine_unified.py	/home/synic	/Projects/exaile/working,	/xl/player
IS-Artist		ertype: Collection		15	engine_normat.py	/home/synic	/Projects/exaile/working	/xt/player
		c in COLLECTIONS:		14	init .py	/home/synic	/Projects/exaile/working	/xl/player
filter="*.py" {		if c.loc_is_member(loc):			radio.py	/home/synic	/Projects/exaile/working	/xl
t py Daughter Da	rling	return c			playlist.py	/home/synic	/Projects/exaile/working,	/xl
ction.py		Inn None		10	common.py	/home/synic	/Projects/exaile/working	
r.py	class O	ollection(trackdb.TrackDB):			nls.py	/home/synic	/Projects/exaile/vorking	/xl
es.py David West					migration.py	/home/synic	/Projects/exaile/working	/xl
nic.py De Phazz		Manages a persistent track database.			transcoder.py	/home/synic	/Projects/exaile/working,	/xl
.py		:param aros: see liclass: xl.trackdb.		atist	trackdb.pv	/home/synic	/Projects/exaile/working	
s.py Dervision								02
PY 🔒 Dead Calm								03
y A Death Cab Fi		8 Forever After (feat. SU						03
tion.py		>>> c = Collection("Test Collection"						0.2
ns.py The Decemb		>>> c.add_library(Library(*./tests/d						03
ders.pypeekine & b		>>> c.rescan_libraries()e (feat. Leigi /		~ ileri				0.3
.py		>>> tracks = list(c.search(artist=	TestArtist ())	19:18ute	xplorerii-i	CNUL Colora Fall	Rublic Lacance as public	unix 4,1 Bot
Cov Deepsky		5	LAdite	# the Fr	ee Software Foundati	ion; either	version 2, or (at your or	otion)
db.py Deepsky & 1			0.2.99.3+	# any la				03
scoder.py	4.4			# There		51.44	Electropic, annihistoria 20	63
by IS DV	der	alobal COLLECTIONS	pickle_attrs=[]): Exame	# Inis p	THOUT ANY WARPANTY	id in the no	pe that it will be useru	
oragna keepeche Mo		self.libraries = dict()	© 2009	# MERCHA	NTABILITY or FITNESS	S FOR A PART	ICULAR PURPOSE. See the	406
r=player_filter="		selfscanning = False	http://www.evals.org	# GNU Ger				06
hitpy		selfscan_stopped = False		#			- Chill Company 1 (2012) - 1	
e.py		self_running_count = 0		# along	with this program: i	if not, write	e to the Free Software	ense
ne unified.pyheel		self. frozen = Falsetrination		# Foundat	tion. Inc., 675 Mass	s Ave, Cambr	idges MAr02139, USA. 20	06
. py & Dilliment		self_libraries_dirty = False						
le.py		pickle attrs += serial libraries	lection	from XL.	nls import gettext a	as		00
ata O DI River / VA		pickle attrs=pickle attrs)	Nuages du Monde	import 1	ogging			06
ctsunix 189,1 Top		COLLECTIONS.add(self)		logger =	logging.getLogger(name_)_		06
ion.py (/home/syn								100
lection	der	freeze_Librargesteeting instant		def	oviderManager(doject	4:06		06
ocessEvent		Prevents "libraries modified" ev	rents from being sent from individual		self.services = {}			06
otifyEventProcesso		add and remove library calls.						000
InotifyNotSupporte			Nuages du Monde	Ddefi	register_provider(se	elf, service	name, provider): 20	06
ansferQueue		thay libraries when you are done	this sends a single event if the		III Registers a proj			006
P O Drew Latta		libraries were modified.	, this sends a single event in the		Negisters o pro-			
								1
nit [Collection]		selffrozen = True			Oparam servicena	one: the name	e of the service [string]	
eze libraries (collin	def	they libraries(self):			Oparam provider:	the object	that is the provider [o]	hiert]
library [Collect]					-1-7 7 3:53			ojecej
ove_library [Coll]		Re-allow "libraries_modified" ev	ents from being sent from individual		if servicename not i	in self.serv	ices:	
p_scan [Collectio]		add and remove library calls. Als	so sends a "libraries_modified"		self.services[se	ervicename]	= 034 showing, 4205 in co	lection
can libraries [Col		freeze libraries.	Modified Since the tast call to		longer.debug(("Pro	vider %(prov	ider)s registered for se	rvice %(service)s
ogress_update [Co]					['provider'	: provider.	name, 'service' : service	ename})
[-]unix 5,5 4%	1:~/Pro:	jects/exaile/working/xl/collection.py			jects/exaile/working	g/xl/provide	rs.py	unix 17,31 3%

Professional Programming IDEs

MBED (ONLINE)

← → C ≜ https://mbed.c	ng/compiler/				습 🔒 📕 역
0) C II W II C 4	ŧ 📇 🚨 M				
New Dimport Workspace	Save Save Al M Comple @ Commit	🕜 Revisions 🖙 🖙 🚧	🗞 🔨 Format		mbed NOP LPC 1768
Program Workspace	main.cop				() Expand
 ○ Mor My Programs ○ An Anti-Conference ○ Anti-Confere	<pre>ipinclude "sbed.b" DigitalOut myled(EDU); if tat main() { wyled = 1; myled = 1; myled = 0; myled = 0; ij ij ij</pre>				× •
	Compiler Output for Program: New				Errors: 0 Warnings: 0 Infos: 0
	Description	ErrNo Resource	In Folder	Location	
Ready.					In 1 col 1 13 1945

INTEL[®] XDK

a Intel® XDK IoT Edition							
PROJECTS Blink	DEVELOP	COLOR DO42					
	· · · · · · · · · · · · · · · · · · ·	; Q, 1.					
Working Files 🔅	File Edit Find View Navigate Help						
main.js Blink node_modules icon.png main.js package.json README.md	 /* A simple node.js application intended to blink the onboard LED on the Intel based development boards such as the Intel(R) Galileo and Edison with Arduino breakout board. MRAA - Low Level Skeleton Library for Communication on CMU/Linux platforms Library in C/C++ to interface with Galileo & other Intel platforms, in a structured and sane API with port nanmes/numbering that match boards & with bindings to javascript & python. The intent is to make it easier for developers and sensor manufacturers to map their sensors & actuators on top of supported hardware and to allow control of low level communication protocol by high level languages & constructs. 						
Search for a service Q. ◆ ▶ Explore Services ①	9 10 var mraa = require('mraa'); //require mraa 11 console.log('MRAA Version: ' + mraa.getVersion()); //write the mraa version to the Intel console	XDK					
Saved Service Bindings	12 13 var myOnboardLed = new mraa.Gpio(13); //LED hooked up to digital pin 13 (or built in pin on Galileo Genl & Gen2)						
+ Create A New Web Service	<pre>14 myUnboardLed.or(mras.JLR_UUI); //set the gpio direction to output 15 var ledState = true; //Boolean to hold the state of Led 16 17 periodicActivity(); //call the periodicActivity function 18 19 function periodicActivity() 20 {</pre>						
	Line 10, Column 43 – 24 Lines INS JavaScript 🥼	Spaces: 4					
IoT Device: edison (192.16 	8.0.123:68888)						

Data Acquisition and Analysis Tools

XIVELY



MICROSOFT AZURE



Visualisation Tools

Grafana Dashboard



NodeRed UI



Summary / Recap of Main Points





IoT refers to a network of interconnected physical devices, vehicles, appliances, and other objects embedded with sensors, software, and network connectivity that enables them to collect and exchange data.



IoT allows for the seamless integration of the physical world with the digital world, creating a network of devices that can interact and collaborate to automate tasks, gather and analyze data, and improve efficiency and convenience in various domains.



By leveraging sensors and connectivity, IoT devices can collect data from their surroundings, monitor their own performance, and respond to commands or triggers. This data can then be analyzed to gain insights, make informed decisions, and optimize processes.

Review Questions



- What is the concept of IoT? How does it relate to the network of interconnected physical objects and data exchange over the Internet?
- What are the key components of IoT hardware? Briefly describe the role of each of the following: sensors, actuators, microcontrollers, and microprocessors.
- How do prototyping tools support IoT development? Provide examples of prototyping tools used for creating IoT prototypes and proofs-of-concept.
- What are data acquisition, analysis, and visualization tools in the context of IoT? How do they contribute to collecting, analyzing, representing, and interpreting IoT data? Give examples of graphical interfaces and dashboards used for IoT data visualization.

What To Expect Next Week



In Class

Preparation for Class