未來之星-智慧科技夏令營

隨著資訊科技進步，未來世界已來臨!為培養下一個世代人才，本營隊由產官學攜手合作，依據高中生的智慧科技學習需求，規劃整合型營隊，以期參與的高中生獲得未來需要多元科技能力，包括硬體與軟體之整合經驗、從操作中學習的實作體驗，以及智慧科技的重要觀念與應用。簡言之，這是一個融合電機、電子、程式、人工智慧、動手實作，兼具理論與實作的高中生終極夏令營!

壹、目的

一、 培育學生由實作中探索學習科技融入生活的重要觀念與應用

二、 提升學生對人工智慧、機器學習等智慧科技能力

三、 鼓勵學生進階研究並參與臺灣國際科學展覽會

貳、主辦與贊助單位:

國立科學教育館、國立臺灣大學電機工程學系　　主辦

聯發科技股份有限公司　　贊助

參、地點:

國立臺灣大學電機二館

肆、時間:

正式課程：108 年 7 月 08 日至 7 月 12 日(週一至週五)

回流輔導：108 年 7 月 20 日(週六)

成果發表：108 年 8 月 03 日(週六)

伍、課表:

時間	7/8 一	時間	7/9 二	7/10 三	7/11 四	7/12 五
9:00-9:20	報到	7:00-8:00	早餐時間	早餐時間	早餐時間	早餐時間
9:20-10:30	開幕典禮	**8:15-9:00**	搭車至臺大	搭車至臺大	搭車至臺大	搭車至臺大
10:30-12:00	專題演講	9:00-12:00	科技創意動手做 2	科技創意動手做 4	人工智慧 1	人工智慧 3
12:00-13:00	午餐	12:00-13:00	午餐	午餐	午餐	午餐
13:00-17:10	科技創意動手做 1	13:00-17:10	科技創意動手做 3	科技創意動手做 4	人工智慧 2	人工智慧 4
17:30	返宿用餐	**17:30**	返宿用餐	返宿用餐	返宿用餐	賦歸

	主題	理論	實作
1.	科技、人工智慧大未來	專題演講:科技、人工智慧大未來	-
2.	科技創意動手做 1 (STEAM Hands-on 1)	理論:微控制器、輸入裝置、輸出裝置原理介紹、軟硬體整合介紹	軟硬體實作基礎
3.	科技創意動手做 2 (STEAM Hands-on 2)	理論:常用電子模組、感測器原理	應用實作:電子模組、感測器應用實作
4.	科技創意動手做 3 (STEAM Hands-on 3)	理論:無線通訊 (Wireless Communication)、網路伺服器 (Network Server)	應用實作:家庭氣象站 (Home Weather Station)
5.	科技創意動手做 4 (STEAM Hands-on 4)	理論:資料結構(Data Structure)、堆疊(Stack)、中綴/後綴 (Infix/Postfix) 轉換	應用實作:計算機 (Calculator)
6.	人工智慧 1 (Artificial Intelligence 1)	理論:人工智慧入門、窮舉法 (Method of Exhaustion)	應用實作: 1A2B 猜數字遊戲 (Bulls and Cows)
7.	人工智慧 2 (Artificial Intelligence 2)	理論: A* 搜尋演算法, 知情/盲目 (informed/uninformed)搜尋演算法	應用實作: 8 塊拼圖遊戲 (8-Puzzle)
8.	人工智慧 3	理論:對抗(Adversarial)搜尋法、	應用實作:黑白棋遊戲

	(Artificial Intelligence 3)	最小值最大化(Minimax)搜尋法，啟發函數(Heuristic)設計	(Reversi)
9.	人工智慧 4 (Artificial Intelligence 4)	理論:類神經網路(Artificial Neural Network)、深度學習 (Deep Learning)	應用實作:黑白棋對弈賽 (Reversi Tournament) (電腦 VS 電腦)
10.	創意提案 (Innovative Proposal)	NABC 創意提案	
11.	回流	專題提案與討論	
12.	成果報告	專題提案與發表	

時間	7/20 六	時間	8/3 六
9:00-12:00	回流	9:00-12:00	成果發表
12:00-13:30	午餐	12:00-13:30	午餐
13:30-17:30	回流	13:30-15:30	成果發表
17:30	賦歸	15:30-17:30	閉幕典禮
		17:30	賦歸

六、講師與助教

林晃巖 教授 台灣大學電機工程學系

黃定洧 教授 台灣大學電機工程學系

李宏毅 教授 台灣大學電機工程學系

王鼎中 老師 台北市立建國高級中學

洪國曉 台灣大學電機工程學系

姚均嚴 台灣大學電機工程學系

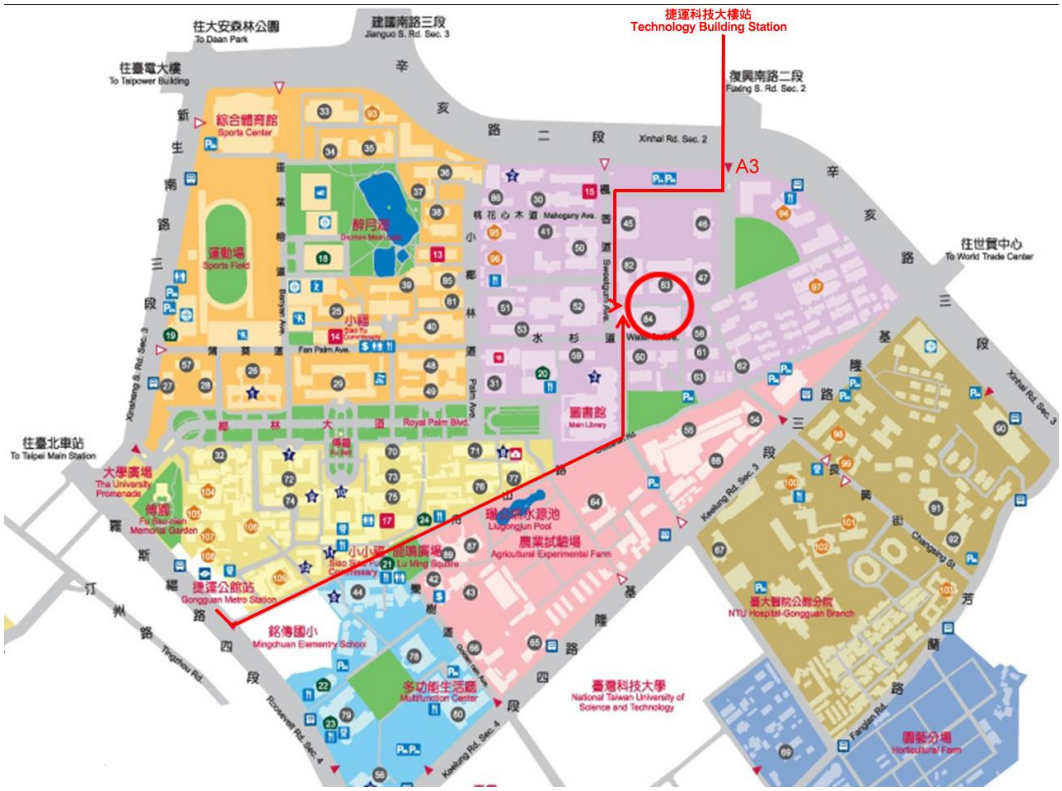
鄭人豪 台灣大學電機工程學系

柒、學員名單

姓名	電子郵件	校名	年級
洪軾凱	sky9109@gmail.com	國立台灣師範大學附屬高級中學	高一
羅元鴻	jhsnu23817@gmail.com	師大附中	高一
劉承翰	liu91hank@gmail.com	國立臺灣師範大學附屬高級中學	高一
林曜儀	narutoyylin@gmail.com	市立大同高中	高一
呂昊謙	howardimportant@gmail.com	桃園市立武陵高中	高一
李宗祐	andy900598@gmail.com	成功高中	高一
吳嘉桐	a91082900@gmail.com	臺北市立建國高級中學	高二
謝秉嶧	37910938casper@gmail.com	台南一中	高一
陳亨翰	v196888888@icloud.com	國立新竹高級中學	高一
陳杰彤	emmatonton@gmail.com	北一女中/Moapa Valley High School	高二
沈倢妤	x0180368905@gmail.com	延平中學	高一
李緒成	peterxcli@gmail.com	濱江國中	國三
張育碩	samchang20030303@gmail.com	成功高中	高一
李嶧	a26195828963@gmail.com	國立羅東高級工業職業學校	高一
方詔陽	johnfang@kimo.com	師大附中	高二
李欣霖	f131391238@gmail.com	成功高中	高一
王彥蓁	jasper01011010@gmail.com	台北市立建國中學	高二

梁育誠	loserda12@gmail.com	建國中學	高二
郭品毅	kuokuoyiyi@gmail.com	臺北市私立東山高級中學	高二
林詠騏	lineric102@gmail.com	政大附中	高一
高惇譯	anaukao910624@gmail.com	大安高工	高二
吳宗翰	gtr0206edwin@gmail.com	建國中學	高二
王少甫	tommywang0807@gmail.com	成功高中	高一
陳彥宇	andy.chen0915@gmail.com	師大附中	高一
陳 曦	evelyn.chen1007@gmail.com	高雄女中	高二
陳柏翰	percy097189@livemail.tw	師大附中	高二
吳仲凱	kte4242@gmail.com	臺北市立建國高級中學	高二
林浩丞	ellelin1963@gmail.com	國立臺灣師範大學附屬高級中學	高一
林君豪	howardlin2002@gmail.com	建國中學	高二
張瑋宸	asgart0916@gmail.com	中大壢中	高一
朱佑中	kevin20030426@gmail.com	建國高中	高一
劉益嘉	hsuypmom@gmail.com	建國中學	高二
任霆	tony20011003@gmail.com	大安高工	高二
林育萱	yusyuanlin0910@gmail.com	台中女中	高一
周君桓	andy.0618.1216@gmail.com	建國中學	高一

Map of National Taiwan University



Map of Taipei Metro

2019 Young Talent Intelligence Technology Camp

7/8-12, 2019

Department of Electrical Engineering,
National Taiwan University

Taipei, Taiwan

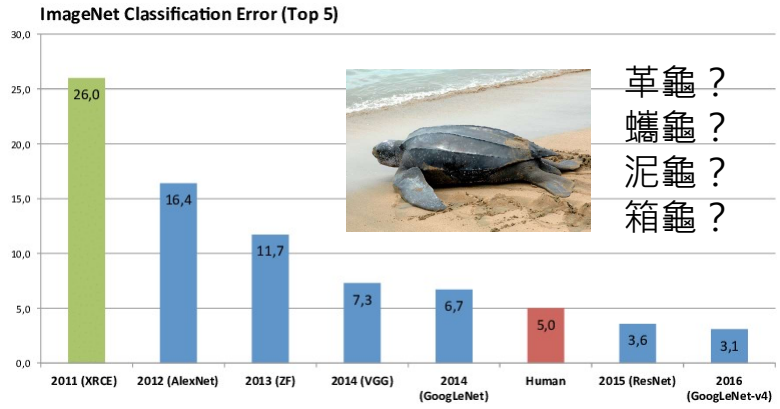
人工智慧技術的現況與未來發展

李宏毅
Hung-yi Lee

語音辨識能力超越人類？

- Microsoft researchers achieve new conversational speech recognition milestone (2016.1
  - Machine 5.9% v.s. Human 5.9%
  - https://www.microsoft.com/en-us/research/blog/microsoft-researchers-achieve-new-conversational-speech-recognition-milestone/
  - Dong Yu, Wayne Xiong, Jasha Droppo, Andreas Stolcke , Guoli Ye, Jinyu Li , Geoffrey Zweig, “Deep Convolutional Neural Networks with Layer-wise Context Expansion and Attention”, Interspeech 2016
- IBM vs Microsoft: 'Human parity' speech recognition record changes hands again (2017.03
  - Machine 5.5% v.s. Human 5.1%
  - http://www.zdnet.com/article/ibm-vs-microsoft-human-parity-speech-recognition-record-changes-hands-again/
  - George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, Bergul Roomi, Phil Hall, “English Conversational Telephone Speech Recognition by Humans and Machines”, arXiv preprint, 2017

影像辨識能力超越人類？



Source of image: https://www.researchgate.net/figure/Winner-results-of-the-ImageNet-large-scale-visual-recognition-challenge-LSVRC-of-the_fig7_324476862

閱讀理解能力超越人類？

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

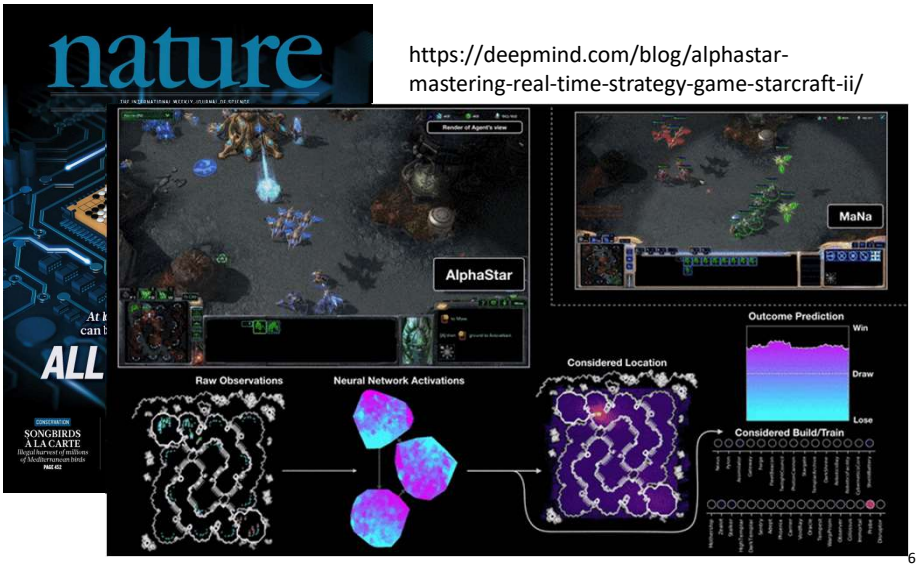
Where do water droplets collide with ice crystals to form precipitation?
within a cloud

SQuAD

https://arxiv.org/pdf/1606.05250.pdf

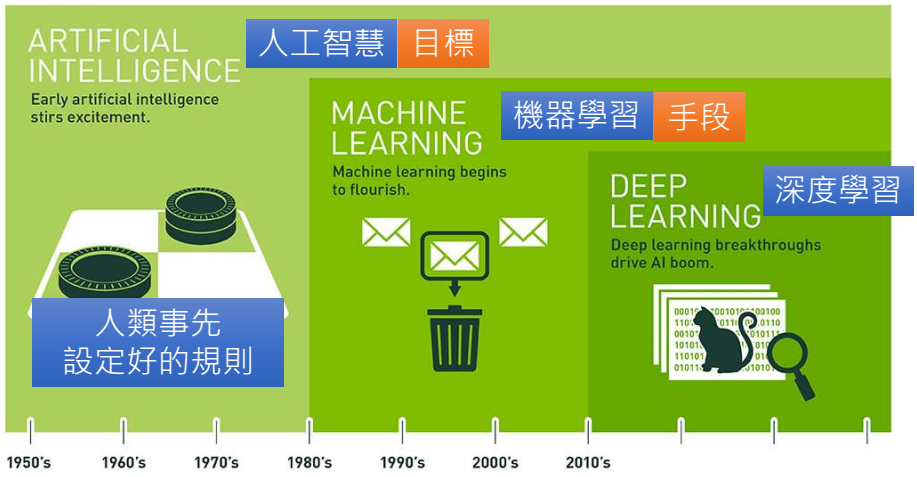
Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433**	**93.160**
2 Oct 05, 2018	BERT (single model) *Google AI Language* https://arxiv.org/abs/1810.04805	85.083	91.835
2 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202
2 Sep 26, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.954	91.677
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490
4 Jul 08, 2018	r-net (ensemble) *Microsoft Research Asia*	84.003	90.147
5 Mar 19, 2018	QANet (ensemble) *Google Brain & CMU*	83.877	89.737

下圍棋、打星海超越人類？



https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/

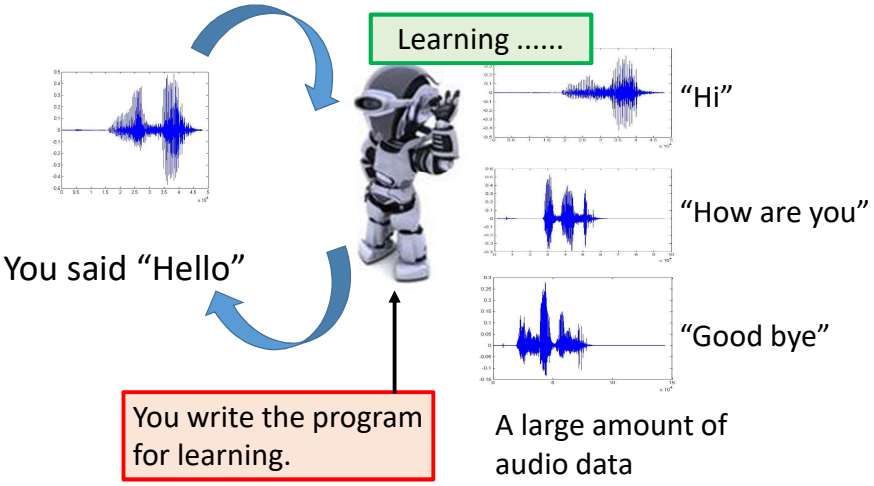
這歸功於機器學習



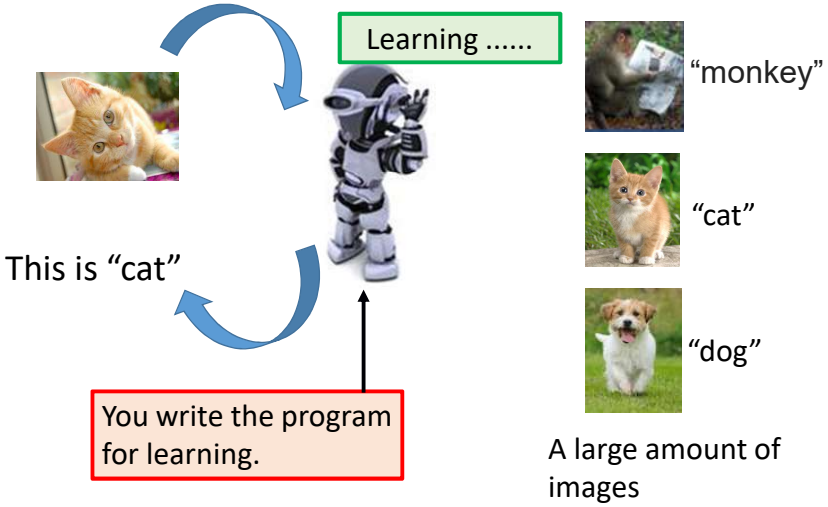
Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Source of image: https://blogs.nvidia.com.tw/2016/07/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

機器學習登場

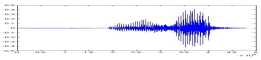


機器學習登場



機器學習
≈ 找一個函數的能力 根據資料

- Speech Recognition

$f($  $)=$ “How are you”

- Image Recognition

$f($  $)=$ “Cat”

- Playing Go

$f($  $)=$ “5-5” (next move)

- Dialogue System

$f($ “How are you?” $)=$ “I am fine.”

(what the user said) (system response)

Regression (回歸)

Regression

The output of the target function f is “scalar”.

Predict PM2.5

Training Data:

Input:

9/01 PM2.5 = 63 9/02 PM2.5 = 65

Output:

9/03 PM2.5 = 100

Input:

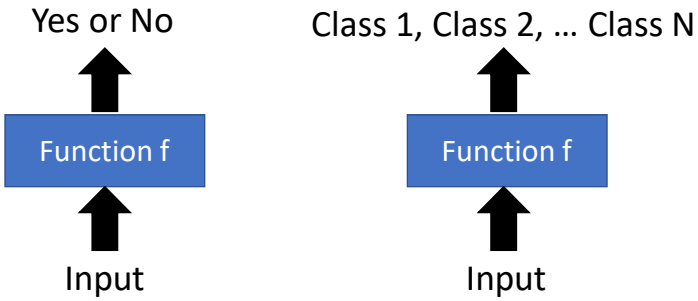
9/12 PM2.5 = 30 9/13 PM2.5 = 25

Output:

9/14 PM2.5 = 20

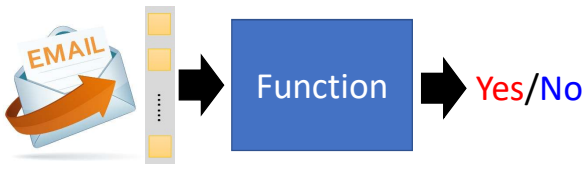
Classification (分類)

- Binary Classification (二元分類)
- Multi-class Classification (多類別分類)



二元分類

Spam filtering

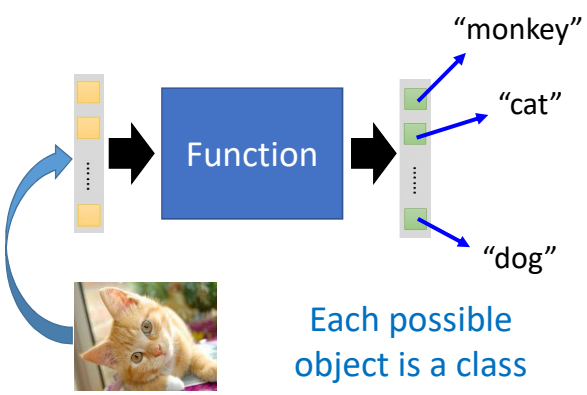


Training Data

(http://spam-filter-review.toptenreviews.com/)

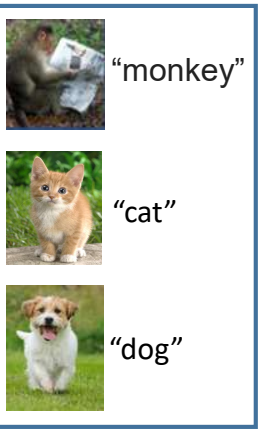
多類別分類

Image Recognition



Each possible object is a class

Training Data



多類別分類

Playing GO

Each position is a class

(19 x 19 classes)

右上星位

左下星位

天元

Next move

Generation (生成)

產生有結構的複雜東西
(例如：文句、圖片)

擬人化的講法—創造

Regression, Classification

機器畫二次元人物 .....

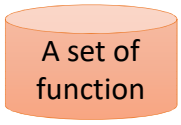
圖片生成：
吳宗翰、謝濬丞、
陳延昊、錢柏均

怎麼自動找一個函式
(機器怎麼做學習)

Framework

Image Recognition:

$f($  $)=$ "cat"



$f_1($  $)=$ "cat"

$f_2($  $)=$ "monkey"

$f_1($  $)=$ "dog"

$f_2($  $)=$ "snake"

Framework

Image Recognition:

$f($  $) =$ "cat"

A set of function

Model

$f_1, f_2 \cdots$

Goodness of function f

Training Data

$f_1($  $) =$ "cat"

$f_2($  $) =$ "monkey"

Better!

$f_1($  $) =$ "dog"

$f_2($  $) =$ "snake"

Supervised Learning (督導式學習)

function input:   

function output: "monkey" "cat" "dog"

Framework

Image Recognition:

$f($  $) =$ "cat"

Step 1

A set of function

Model

$f_1, f_2 \cdots$

Step 2

Goodness of function f

Training Data

Step 3

Pick the "Best" Function

f^*

Training

Testing

"cat"

Using f^*

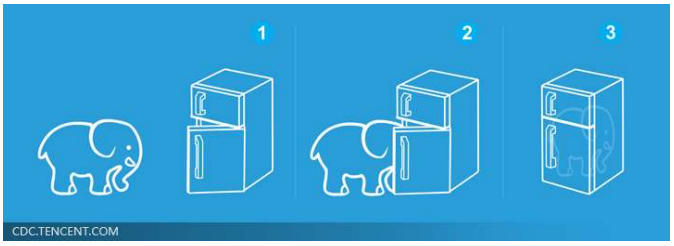
"monkey" "cat" "dog"

機器學習好簡單 .....

Step 0: What kind of function do you want to find?

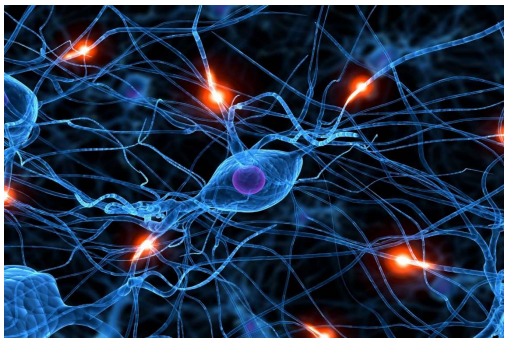
Step 1: define a set of function → Step 2: goodness of function → Step 3: pick the best function

就好像把大象放進冰箱 .....



Deep Learning (深度學習)

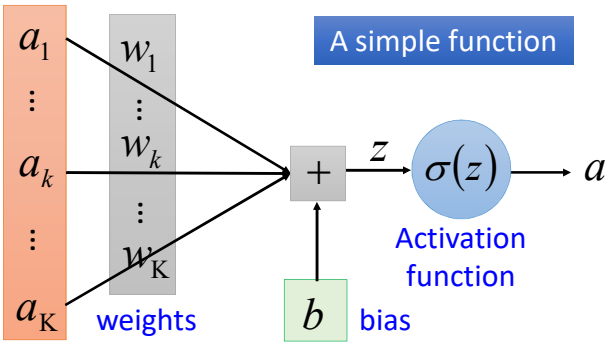
- Deep learning, SVM, decision tree ….
  - →using different ways to represent a function
- Using neural network (神經網路) to represent a function



Neural Network (神經網路)

Neuron (神經元)

$$z = a_1 w_1 + \ldots + a_k w_k + \ldots + a_K w_K + b$$



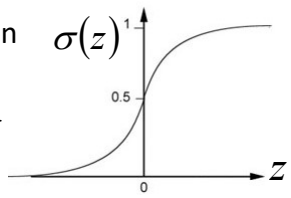
Weights and biases are called network parameters

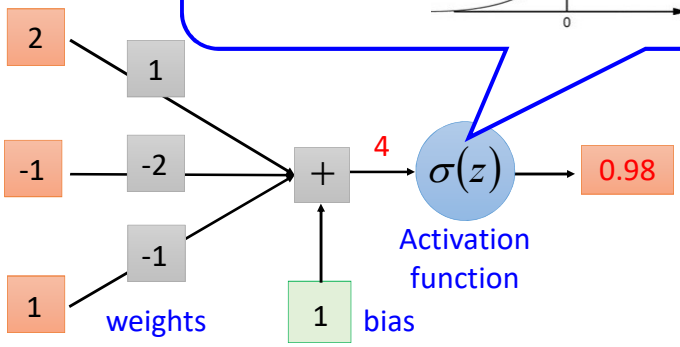
Neural Network (神經網路)

Neuron

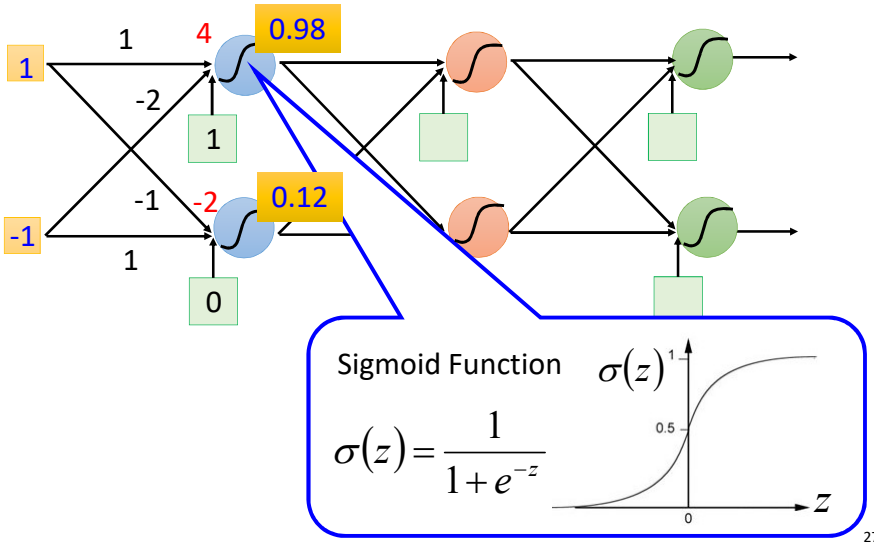
Sigmoid Function

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

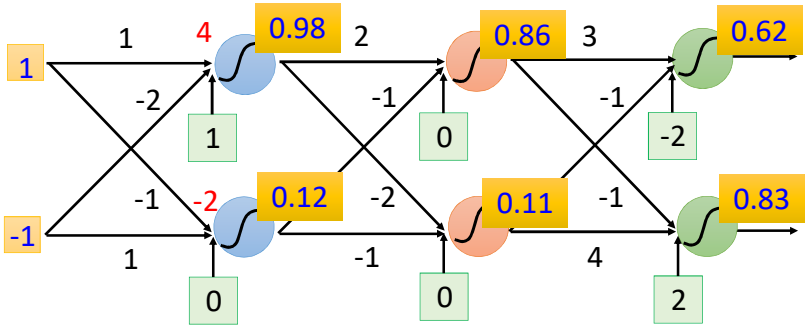




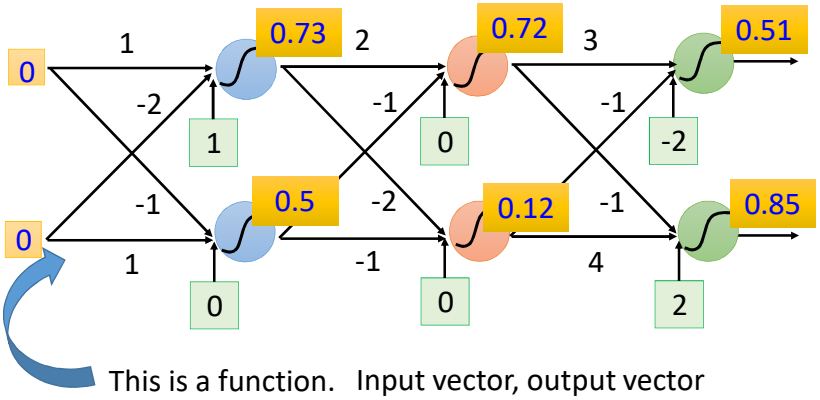
Neural Network (神經網路)



Neural Network (神經網路)



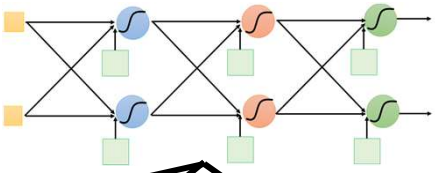
Neural Network (神經網路)



This is a function. Input vector, output vector

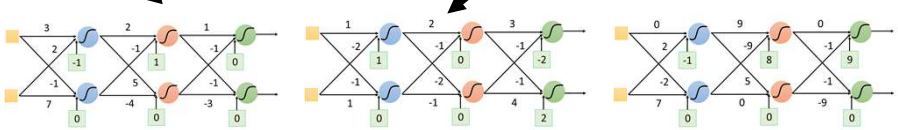
$$f\left(\begin{bmatrix}1\\-1\end{bmatrix}\right)=\begin{bmatrix}0.62\\0.83\end{bmatrix} \quad f\left(\begin{bmatrix}0\\0\end{bmatrix}\right)=\begin{bmatrix}0.51\\0.85\end{bmatrix}$$

function set



人類提供了網路的架構

架構是神經網路的天賦

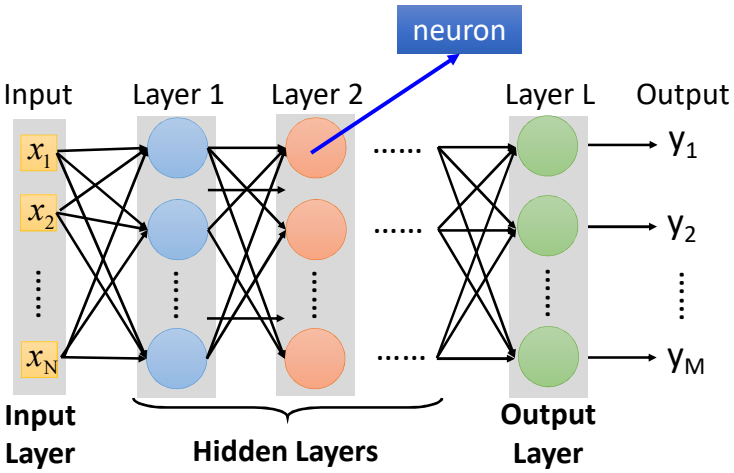


f_1 f_2 f_3

機器自己根據資料找出參數 (也就是選擇了某一個 function)

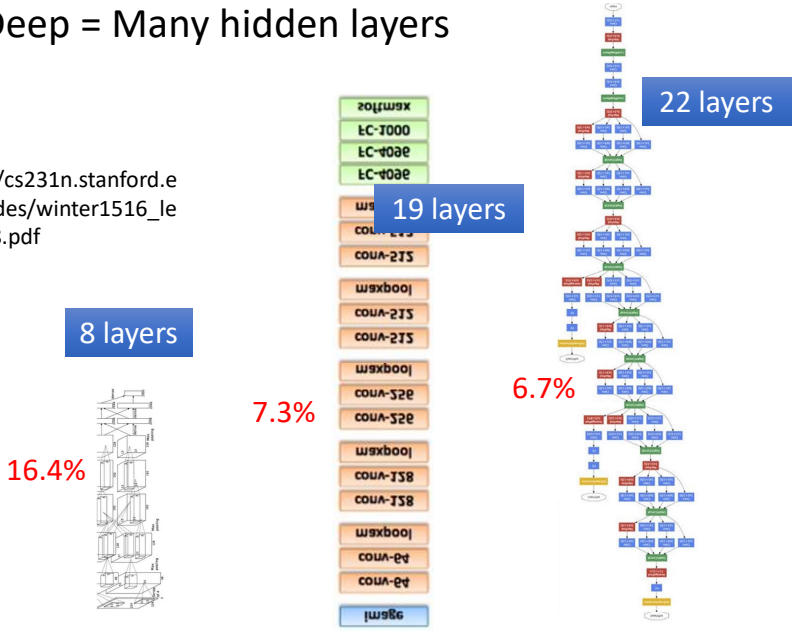
機器自己後天學習的成果

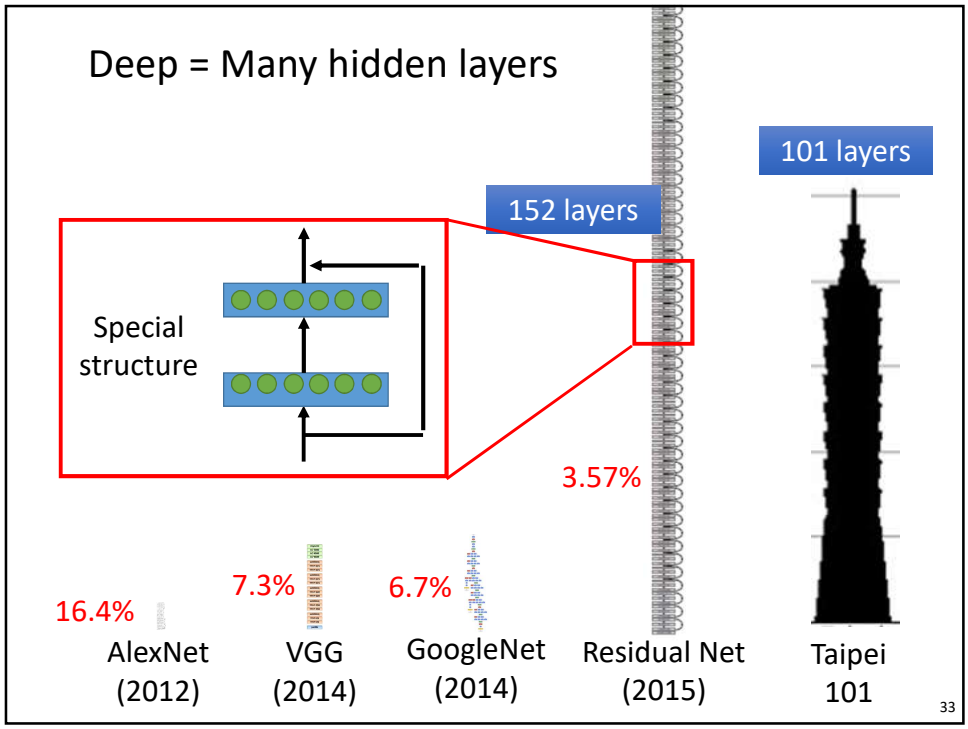
Fully Connected Feedforward Network



Deep = Many hidden layers

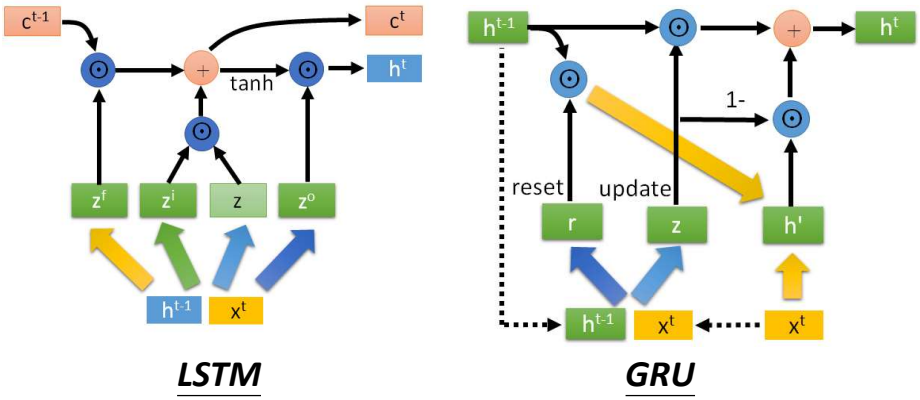
http://cs231n.stanford.edu/slides/winter1516_lecture8.pdf





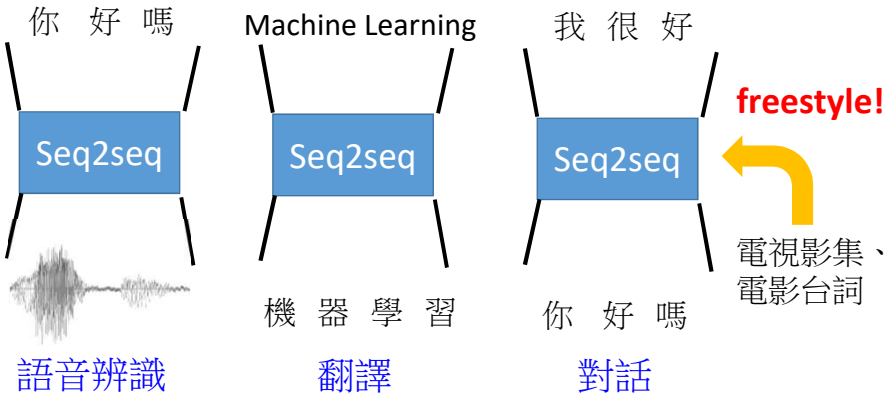
Different Network Structures

- CNN, LSTM, GRU, etc. are just different ways to connect neurons.



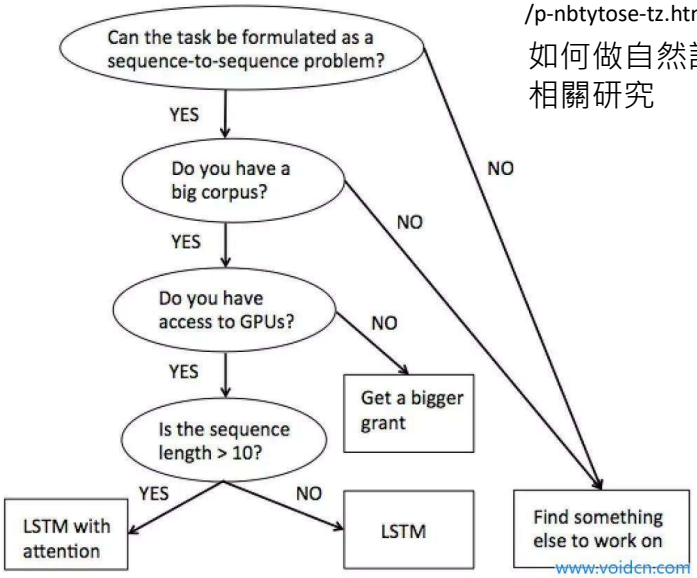
Sequence-to-sequence (Encoder-Decoder Architecture)

- Both input and output are both sequences ***with different lengths***.



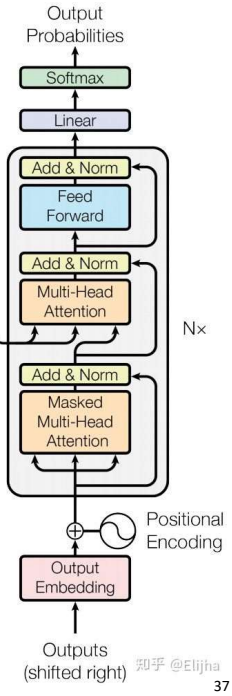
http://www.voidcn.com/article/p-nbtytose-tz.html

如何做自然語言處理相關研究



Sequence-to-sequence

- Transformer

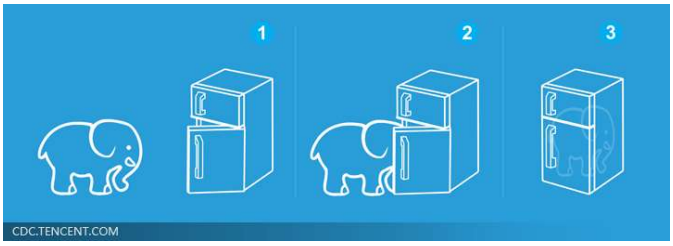


機器學習好簡單 .....

Step 0: What kind of function do you want to find?

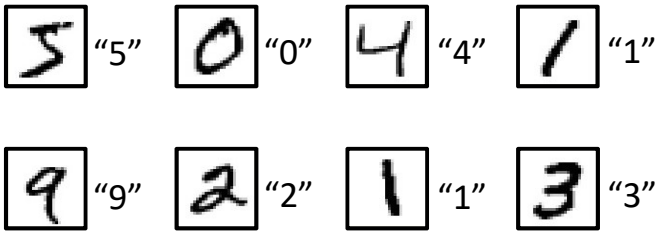
Step 1: define a set of function → Step 2: goodness of function → Step 3: pick the best function

就好像把大象放進冰箱 .....



手寫數字辨識

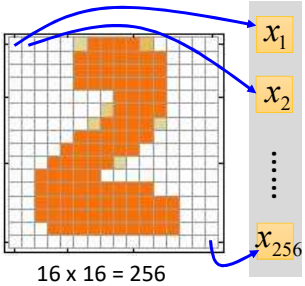
- Training data: images and their labels



The learning target is defined on the training data.

手寫數字辨識

Input



16 x 16 = 256

Ink → 1

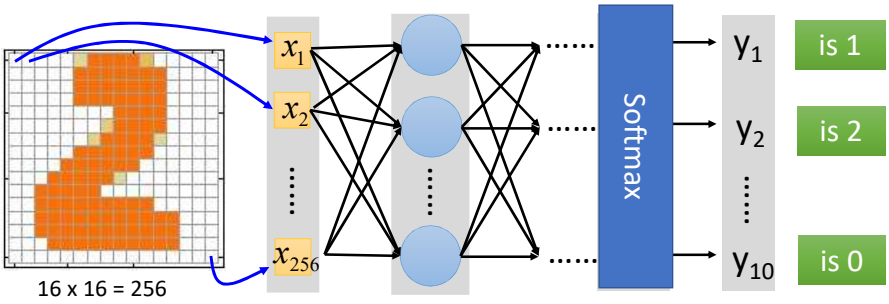
No ink → 0

Output

The image is "2"

Each dimension represents the confidence of a digit.

What is a good function?



16 x 16 = 256

Ink → 1

No ink → 0

A good function should .....

Input:  → y_1 has the maximum value

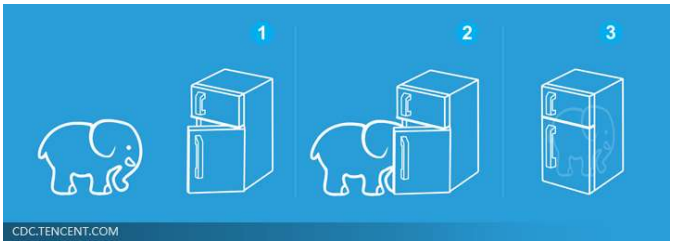
Input:  → y_2 has the maximum value

機器學習好簡單 .....

Step 0: What kind of function do you want to find?

Step 1: define a set of function → Step 2: goodness of function → Step 3: pick the best function

就好像把大象放進冰箱 .....



How to pick the best function

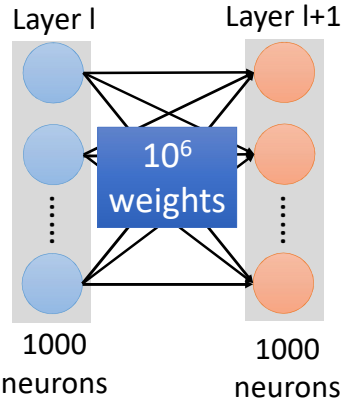
Find ***network parameters*** θ^* that minimize total loss L

Enumerate all possible values

Network parameters $\theta = \{w_1, w_2, \cdots, b_1, b_2, b_3, \cdots\}$

Millions of parameters

E.g. speech recognition: 8 layers and 1000 neurons each layer



Gradient Descent

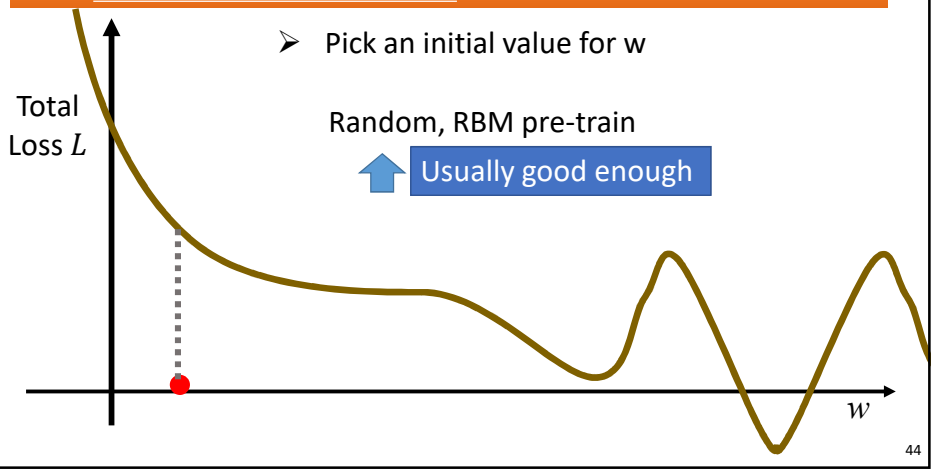
Network parameters $\theta = \{w_1, w_2, \cdots, b_1, b_2, \cdots\}$

Find ***network parameters*** θ^* that minimize total loss L

- Pick an initial value for w

Random, RBM pre-train

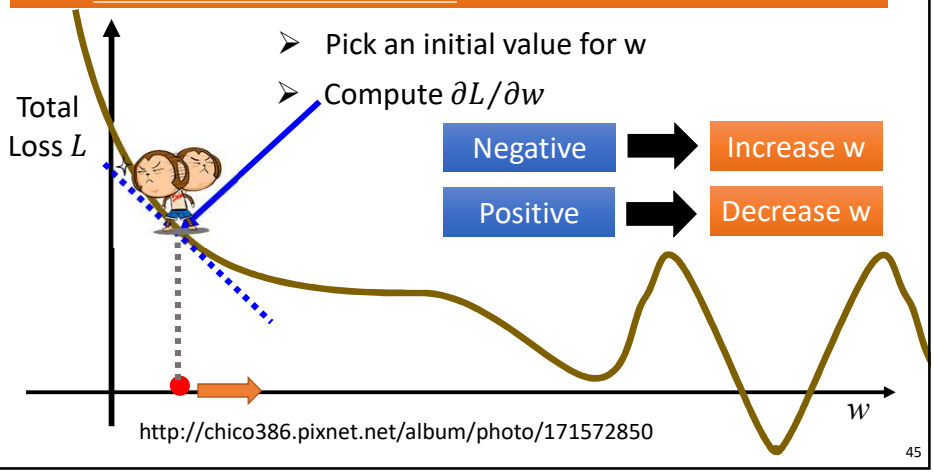
Usually good enough



Gradient Descent

Network parameters $\theta = \{w_1, w_2, \cdots, b_1, b_2, \cdots\}$

Find ***network parameters*** θ^* that minimize total loss L

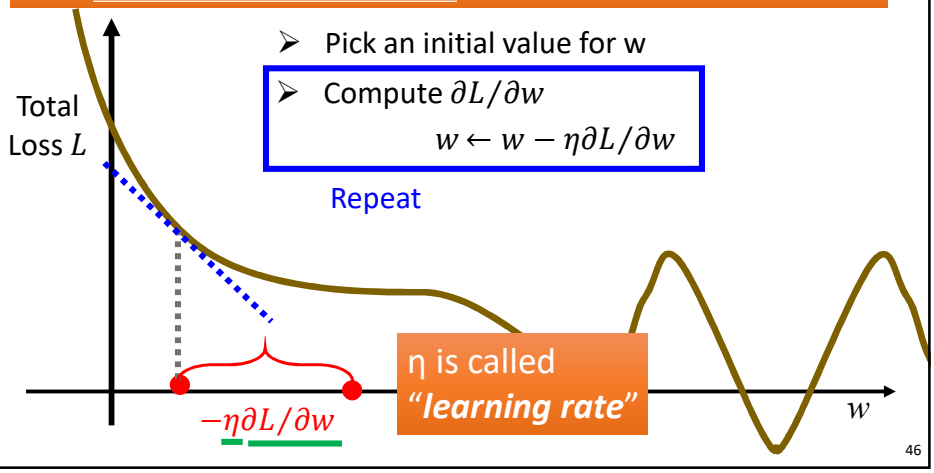


http://chico386.pixnet.net/album/photo/171572850

Gradient Descent

Network parameters $\theta = \{w_1, w_2, \cdots, b_1, b_2, \cdots\}$

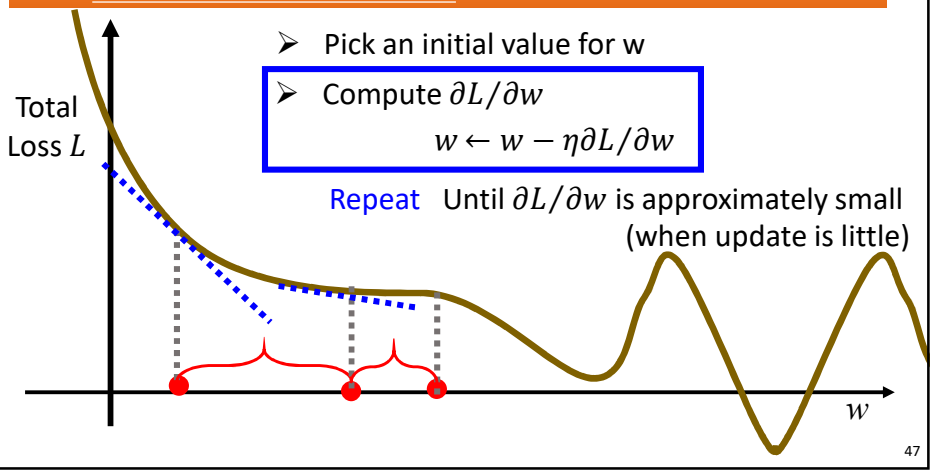
Find ***network parameters*** θ^* that minimize total loss L



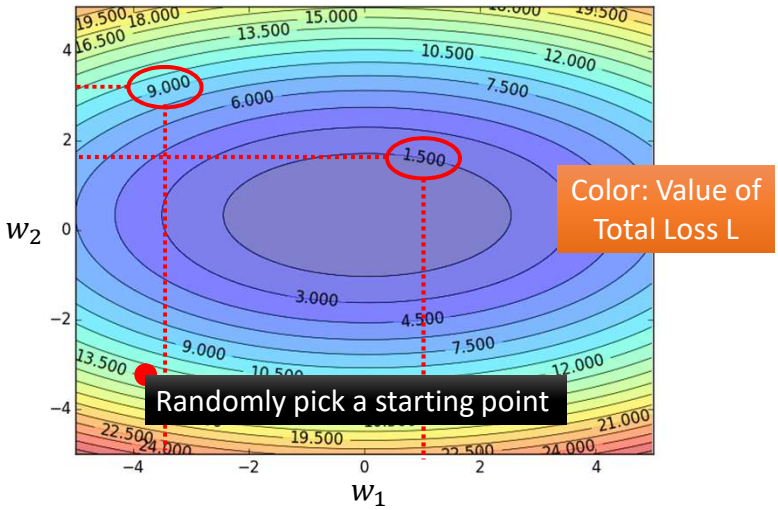
Gradient Descent

Network parameters $\theta = \{w_1, w_2, \cdots, b_1, b_2, \cdots\}$

Find ***network parameters*** θ^* that minimize total loss L

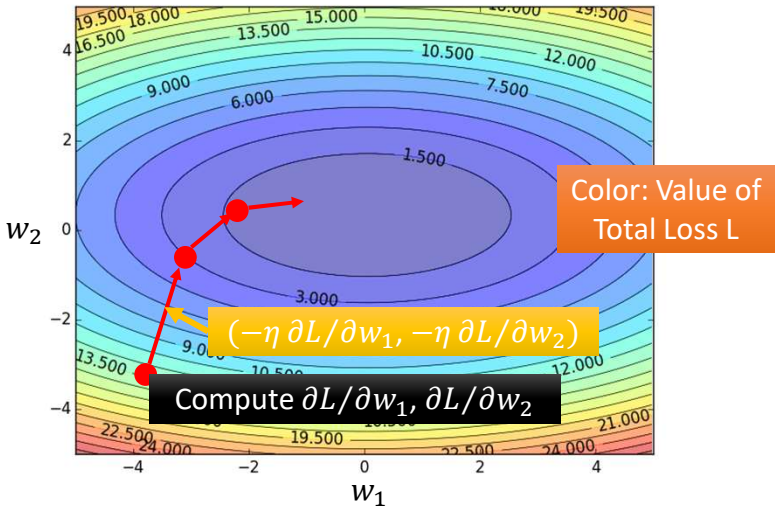


Gradient Descent

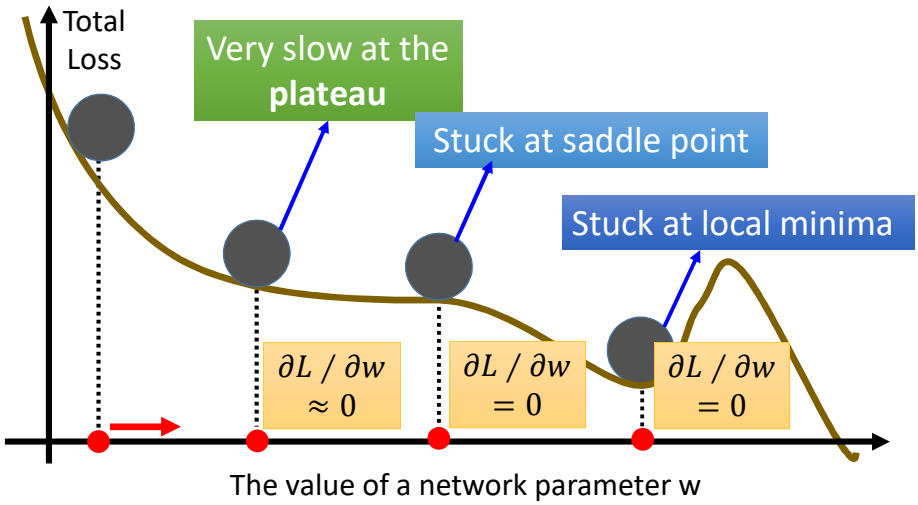


Gradient Descent

Hopfully, we would reach a minima .....



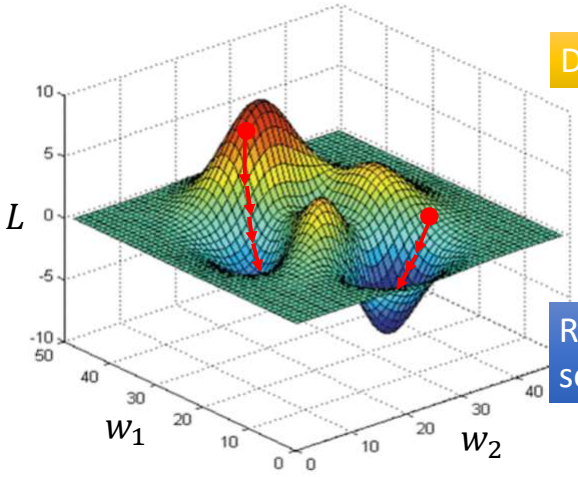
Local Minima



Local Minima

每次訓練結果都不一樣，
有時候確實需要一點運氣 .....

- Gradient descent never guarantee global minima



Different initial point

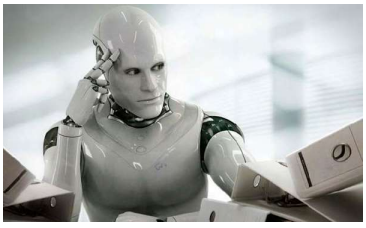
Reach different minima, so different results

Gradient Descent

This is the “learning” of machines in deep learning ……

➡ Even alpha go using this approach.

People image ……



Actually …..

I hope you are not too disappointed :p

Gradient Descent

- Backpropagation: an efficient way to compute $\partial L/\partial w$ in neural network

theano

Caffe

libdnn
台大周伯威
同學開發

Ref: https://www.youtube.com/watch?v=ibJpTrp5mcE

超越督導式學習

Semi-supervised (半督導)

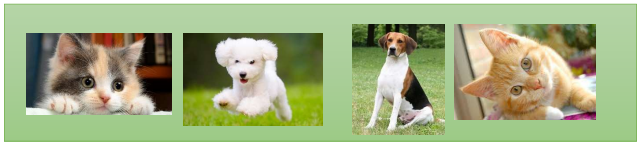
For example, recognizing cats and dogs

Labelled data


cat


dog

Unlabeled data

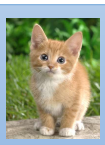


(Images of cats and dogs)

Transfer Learning (遷移學習)

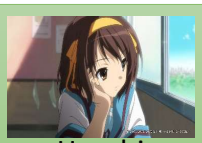
For example, recognizing cats and dogs

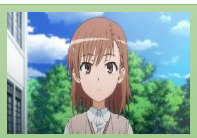
Labelled data


cat


dog


elephant


Haruhi



Data not related to the task considered (can be either labeled or unlabeled)

Reinforcement Learning (增強式學習)

Supervised v.s. Reinforcement

- Supervised:

Next move: "5-5"

Next move: "3-3"

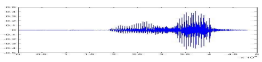
- Reinforcement Learning

First move → …… many moves …… → Win!

Alpha Go is supervised learning + reinforcement learning.

Unsupervised (非督導)

Training AI ***without paired data***



x

Speech Recognition

"How are you"

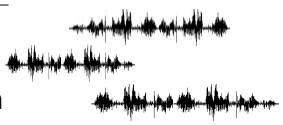
y

Supervised

x_1:  x_2:  x_3: 

y_1: Hello y_2: Good y_3: I am fine

Unsupervised



AI listening in the environment

Audio x

Text y

AI reading documents on the Internet

Unsupervised Learning

It is good.
It's a good day.
I love you.

positive sentences

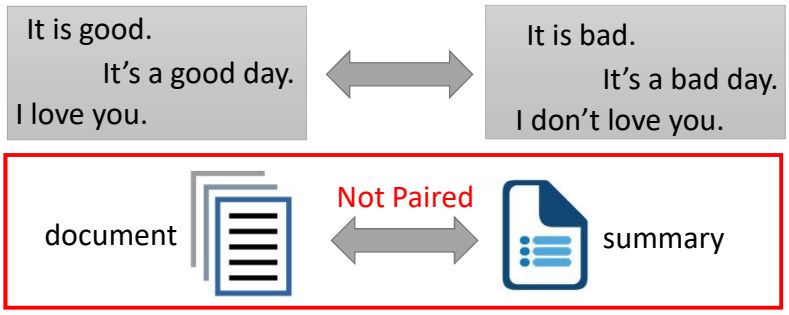
It is bad.
It's a bad day.
I don't love you.

negative sentences

Text Style Transfer

- Negative sentence to positive sentence:
  it's a crappy day → it's a great day
  i wish you could be here → you could be here
  it's not a good idea → it's good idea
  i miss you → i love you
  i don't love you → i love you
  i can't do that → i can do that
  i feel so sad → i happy
  it's a bad day → it's a good day
  it's a dummy day → it's a great day
  sorry for doing such a horrible thing → thanks for doing a great thing
  my doggy is sick → my doggy is my doggy
  my little doggy is sick → my little doggy is my little doggy

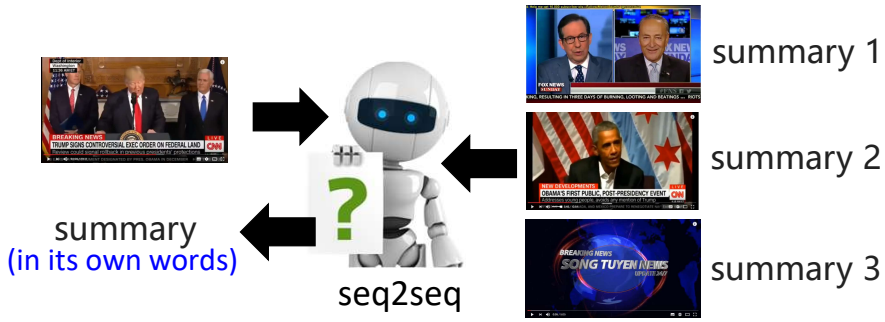
Unsupervised Learning



unsupervised abstractive summarization.

Abstractive Summarization

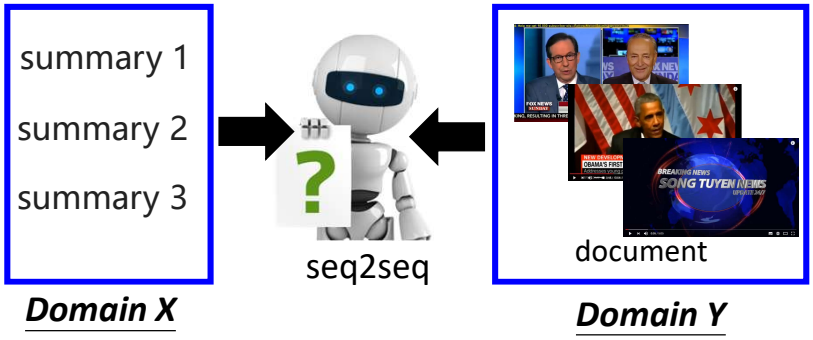
- Now machine can do **abstractive summary** by seq2seq (write summaries in its own words)



Supervised: We need lots of labelled training data.

Unsupervised Abstractive Summarization

- Now machine can do **abstractive summary** by seq2seq (write summaries in its own words)



感謝 王耀賢 同學提供實驗結果

Unsupervised Abstractive Summarization

- ***Document***:澳大利亞今天與13個國家簽署了反興奮劑雙邊協議,旨在加強體育競賽之外的藥品檢查並共享研究成果 ……
- ***Summary***:
  - Human:澳大利亞與13國簽署反興奮劑協議
  - Unsupervised:澳大利亞加強體育競賽之外的藥品檢查
- ***Document***:中華民國奧林匹克委員會今天接到一九九二年冬季奧運會邀請函,由於主席張豐緒目前正在中南美洲進行友好訪問,因此尚未決定是否派隊赴賽 ……
- ***Summary***:
  - Human:一九九二年冬季奧運會函邀我參加
  - Unsupervised:奧委會接獲冬季奧運會邀請函

感謝 王耀賢 同學提供實驗結果

Unsupervised Abstractive Summarization

- ***Document***:據此間媒體27日報道,印度尼西亞蘇門答臘島的兩個省近日來連降暴雨,洪水泛濫導致塌方,到26日為止至少已有60人喪生,100多人失蹤 ……
- ***Summary***:
  - Human:印尼水災造成60人死亡
  - Unsupervised:印尼門洪水泛濫導致塌雨
- ***Document***:安徽省合肥市最近為領導幹部下基層做了新規定:一律輕車簡從,不準搞迎來送往、不準搞層層陪同 ……
- ***Summary***:
  - Human:合肥規定領導幹部下基層活動從簡
  - Unsupervised:合肥領導幹部下基層做搞迎來送往規定:一律簡

Unsupervised Learning

It is good.
It's a good day.
I love you.

It is bad.
It's a bad day.
I don't love you.

document

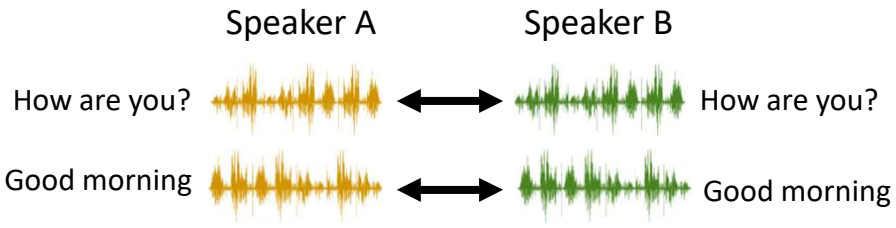
summary

Speaker A

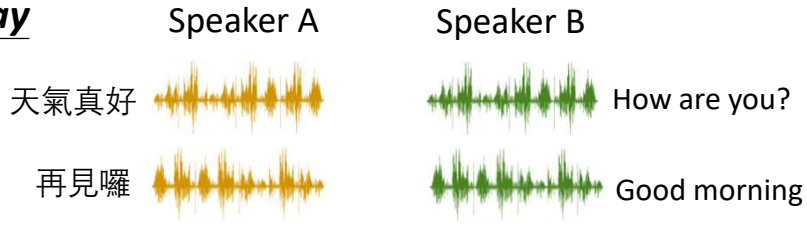
Speaker B

Voice Conversion

In the past



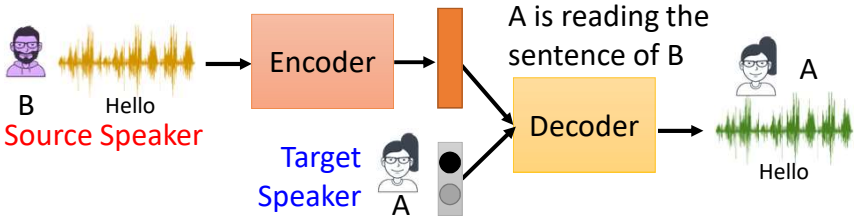
Today



Speakers A and B are talking about completely different things.

69

Demo



Source:

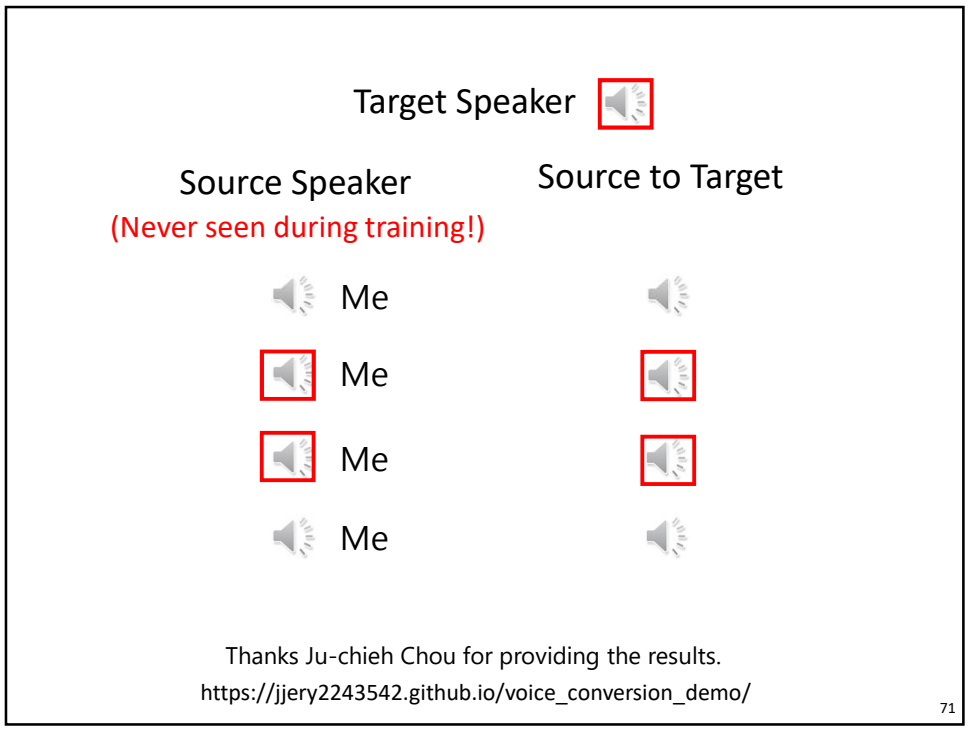
Target:

Source to Target:

Thanks Ju-chieh Chou for providing the results.

https://jjery2243542.github.io/voice_conversion_demo/

70

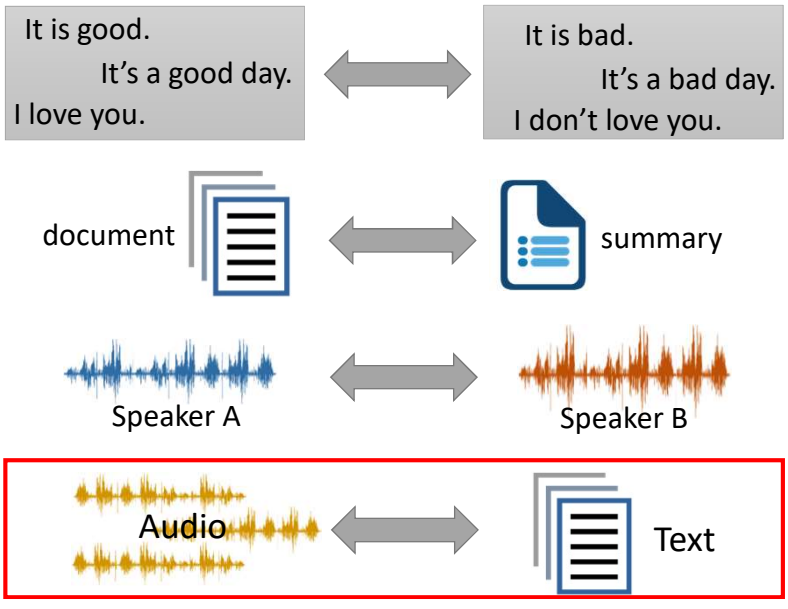


Thanks Ju-chieh Chou for providing the results.

https://jjery2243542.github.io/voice_conversion_demo/

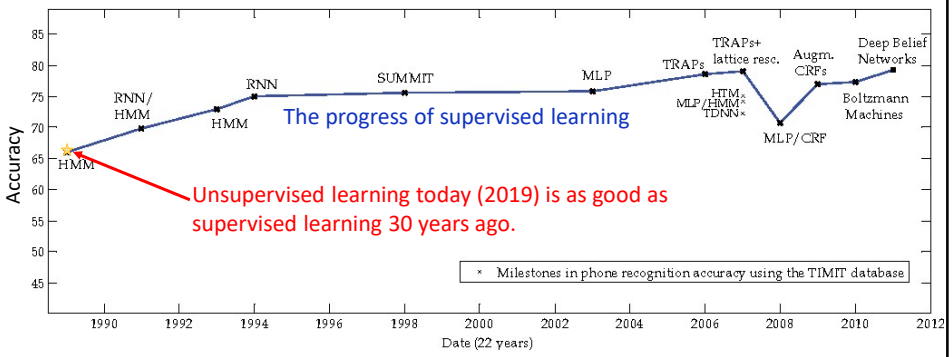
71

Unsupervised Learning



72

Approaches		Matched (all 4000)		Nonmatched (3000/1000)	
		FER	PER	FER	PER
(I) Supervised (labeled)					
(a) RNN Transducer [23]		-	17.7	-	-
(b) standard HMMs		-	21.5	-	-
(c) Phoneme classifier		27.0	28.9	-	-
(II) Unsupervised (with oracle boundaries)					
(d) Relationship mapping GAN [22]		40.5	40.2	43.6	43.4
(e) Segmental Emperical-ODM [23]		33.3	32.5	40.0	40.1
(f) Proposed: GAN		27.6	28.5	32.7	34.3
(III) Completely unsupervised (no label at all)					
(g) Segmental Emperical-ODM [23]		-	36.5	-	41.6
Proposed	iteration 1 (h) GAN	48.3	48.6	50.3	50.0
	iteration 1 (i) GAN/HMM	-	30.7	-	39.5
	iteration 2 (j) GAN	41.0	41.0	44.3	44.3
	iteration 2 (k) GAN/HMM	-	27.0	-	35.5
	iteration 3 (l) GAN	39.7	38.4	45.0	44.2
	iteration 3 (m) GAN/HMM	-	26.1	-	33.1



The image is modified from: Phone recognition on the TIMIT database Lopes, C. and Perdigão, F., 2011. Speech Technologies, Vol 1, pp. 285--302.

機器學習的下一步

說出為什麼「我知道」

說出為什麼「我知道」

- 神馬漢斯

說出為什麼「我知道」

- Goal of ML Explanation ≠ you completely know how the ML model work ➡ Not necessary
  - Some people don't trust deep learning because they think it is a Black Box.
  - Human brain is also a Black Box!
- Goal of ML Explanation is

Make people (your customers, your boss, yourself) comfortable.

讓人覺得高興

Personalized explanation in the future

Interpretable v.s. Powerful

- Some models are intrinsically interpretable.
  - For example, linear model (from weights, you know the importance of features)
  - But ……. not very powerful.
- Deep network is difficult to interpret.

Because deep network is a black box, we don't use it.

削足適履 ☹

- But it is more powerful than linear model …

Let's make deep network explainable.

Saliency Map

https://arxiv.org/abs/1312.6034

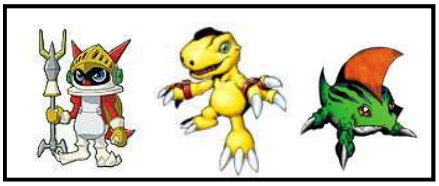
https://medium.com/@tyreeostevenson/teaching-a-computer-to-classify-anime-8c77bc89b881

Task

Pokémon images: https://www.Kaggle.com/kvpratama/pokemon-images-dataset/data

Digimon images: https://github.com/DeathReaper0965/Digimon-Generator-GAN

Pokémon



Digimon

Testing Images:

Experimental Results

```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same', input_shape=(120,120,3)))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(256, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(1024))
model.add(Activation('relu'))
model.add(Dense(2))
model.add(Activation('softmax'))
```

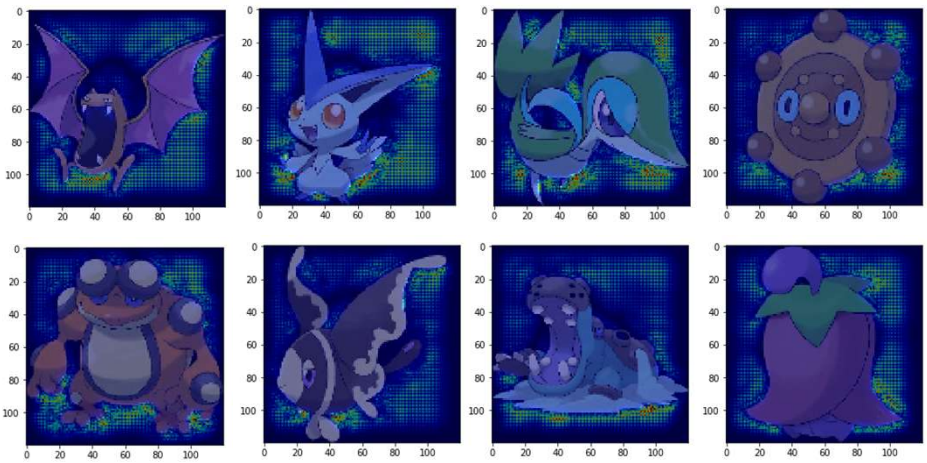
Training Accuracy: 98.9%

Testing Accuracy: 98.4%

太神啦!!!!!!

Saliency Map

Saliency Map



What Happened?

- All the images of Pokémon are PNG, while most images of Digimon are JPEG.

PNG 檔透明背景

讀檔後背景是黑的!

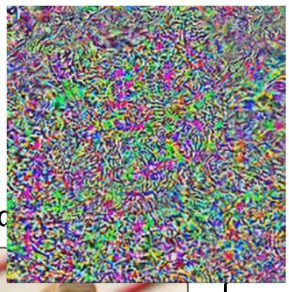
Machine discriminate Pokémon and Digimon based on Background color.

➡ This shows that explainable ML is very critical.

機器學習的下一步

AI 其實很容易被騙

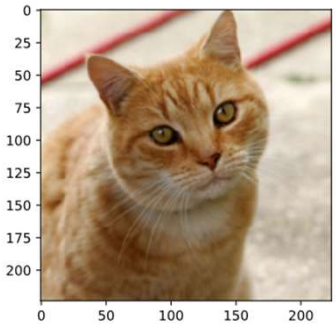
欺騙 AI



=

50x

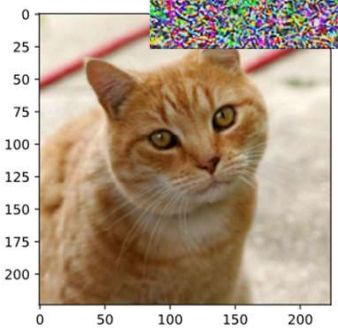
Original Image



Tiger Cat
0.64

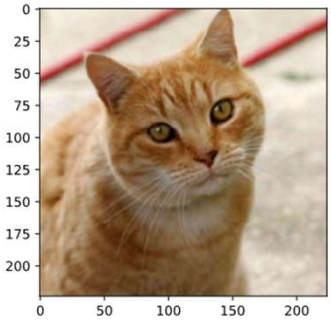
-

Attac

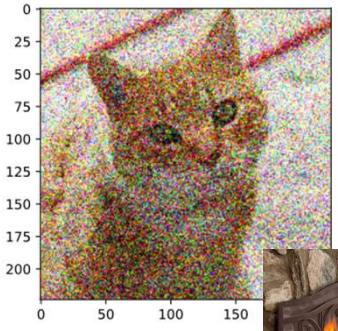


Star Fish
1.00

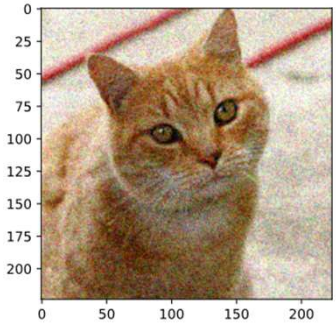
87



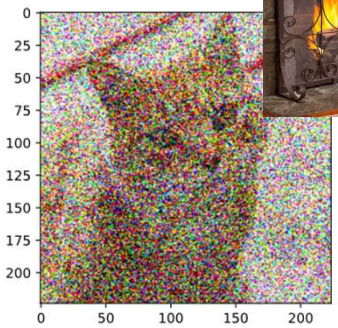
tiger cat



Persian cat



tabby cat

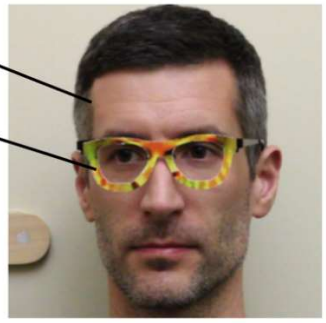


fire screen

88

Distance/Angle	Subtle Poster	Subtle Poster Right Turn	Camouflage Graffiti	Camouflage Art (LISA-CNN)	Camouflage Art (GTSRB-CNN)
5′ 0°					
5′ 15°					
10′ 0°					
10′ 30°					
40′ 0°					
Targeted-Attack Success	100%	73.33%	66.67%	100%	80%

https://arxiv.org/abs/1707.08945



新的資安問題

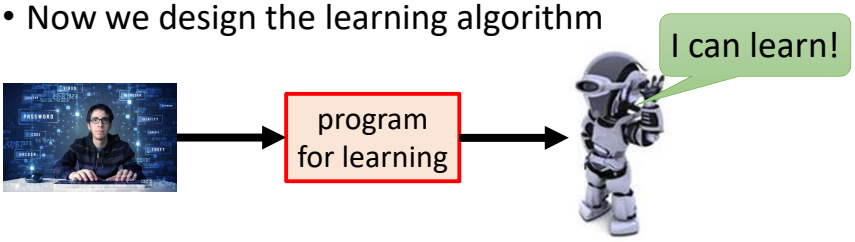
https://arxiv.org/abs/1801.00349

機器學習的下一步

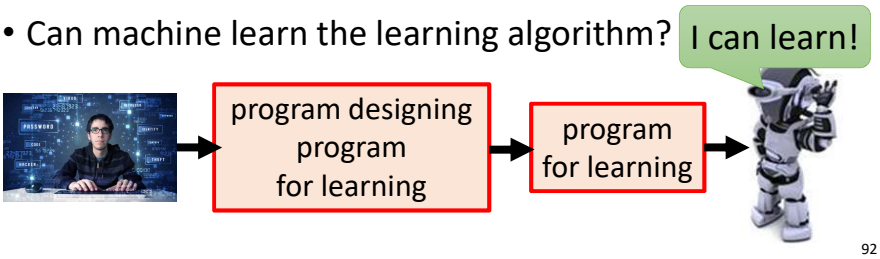
學習如何學習

學習如何學習

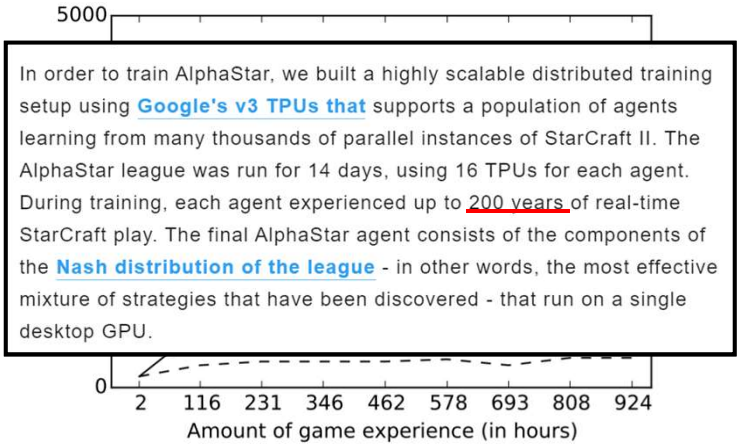
- Now we design the learning algorithm



- Can machine learn the learning algorithm?



機器其實很笨 ...



http://web.stanford.edu/class/psych209/Readings/LakeEtAlBBS.pdf

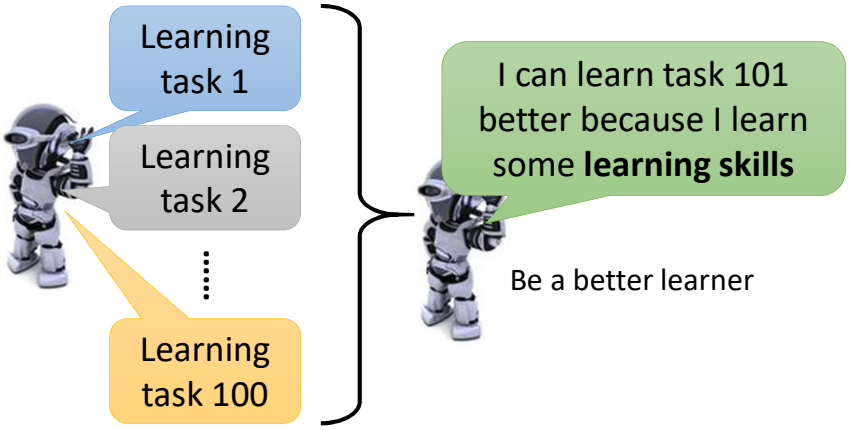
一定需要很多訓練資料嗎？

- Few-shot learning

學習如何學習

Task 1: speech recognition
Task 2: image recognition
⋮
Task 100: text classification

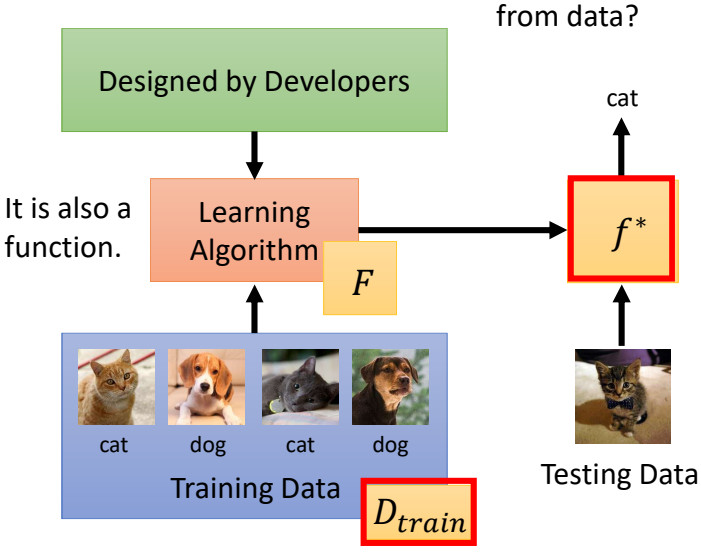
- Meta learning = Learn to learn



Meta Learning

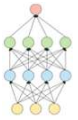
$f^* = F(D_{train})$

Can machine find F from data?



Meta Learning

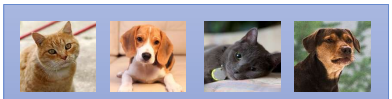
Machine Learning ≈ 根據資料找一個函數 f 的能力

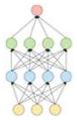


$f($  $) =$ "Cat"

Meta Learning

≈ 根據資料找一個找一個函數 f 的函數 F 的能力

$F($ 

$) = f^*$ 

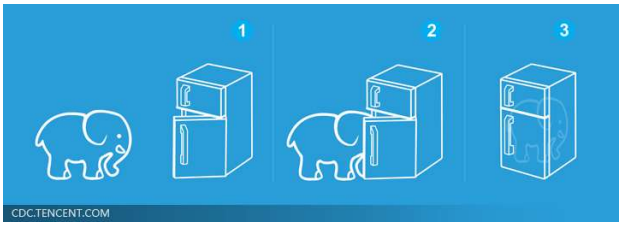
~~Machine~~ Learning is Simple

Meta

Step 1: define a set of ~~function~~ → Step 2: goodness of ~~function~~ → Step 3: pick the best ~~function~~

Function f ➡ Learning algorithm F

就好像把大象放進冰箱 .....



Face Verification

In each task:

Few-shot Learning

https://support.apple.com/zh-tw/HT208109

To Learn More …..

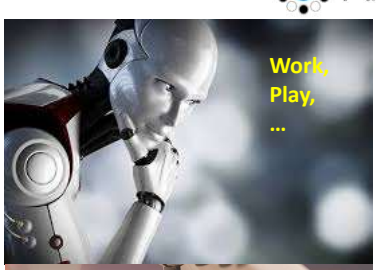
(YouTube Channel of Machine Learning, 37K subscribers, 3M total views)

Course Introduction

Ideal vs. Current AI


AlphaGo

Google Dee
2014~



Computer plays,
Men work!

Equations and Interpretations

From **Data Mining**,

$$man = eat + sleep + play + work$$
$$pig = eat + sleep$$

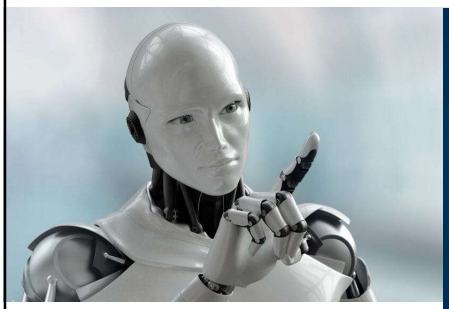
From **AI**,

$$man - pig = play + work$$
$$man - play = pig + work$$

Conclusion: Man can't play = Pig can only work

It's a joke!
However, interpretation may not be easy for AI.
joke may not be appreciated by AI.
we hope both man and AI can work and play.

Dream or Nightmare?



Game Boy

It plays,
Men play!

GAME BOY

Nintendo
1989~

Hardware
+
Software

The BetaGo Project

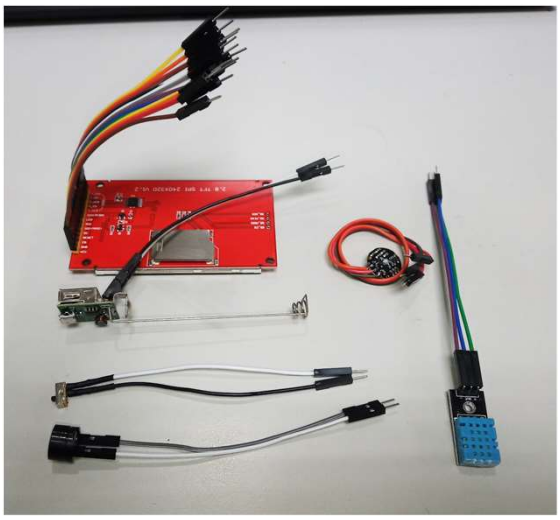
http://140.112.17.252/wordpress/

Objective

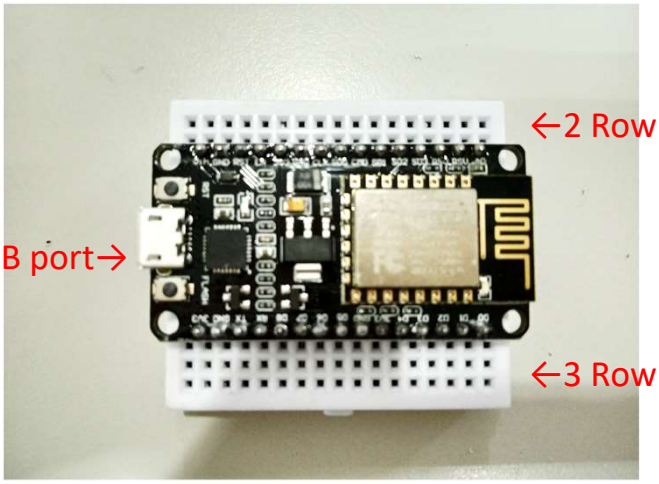
- A Game Baby
  - MCU-based
  - Programs
- Learn how it works

Assembly Instructions

Connect wire to all device

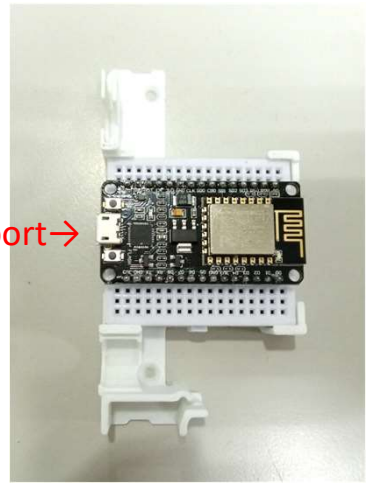


Place ESP8266 on Bread Board



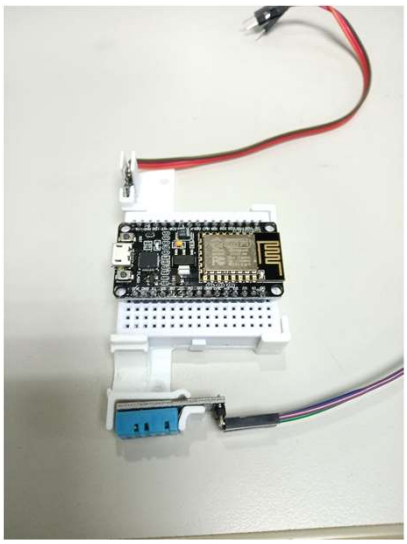
Paste Bread Board to the T Component

111

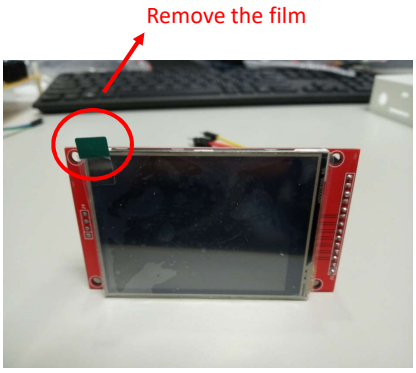


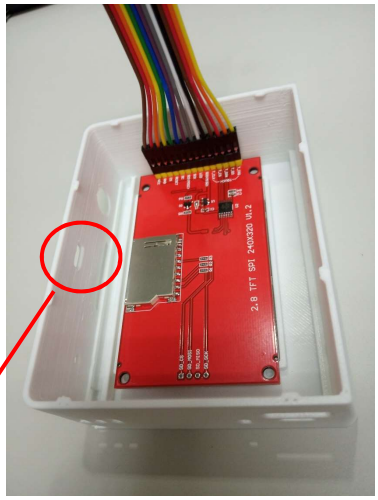
Placed Humidity Sensor & Heartbeat Sensor on the T Component

112

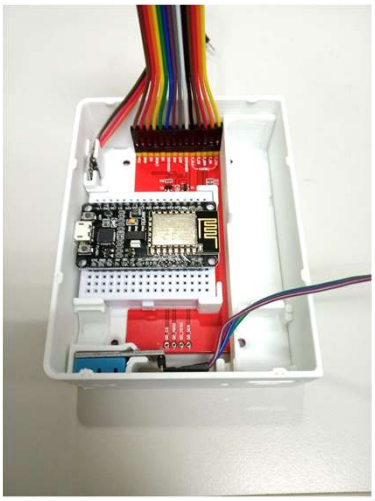


Place Touch Screen into the Case





Place T Component into the Case

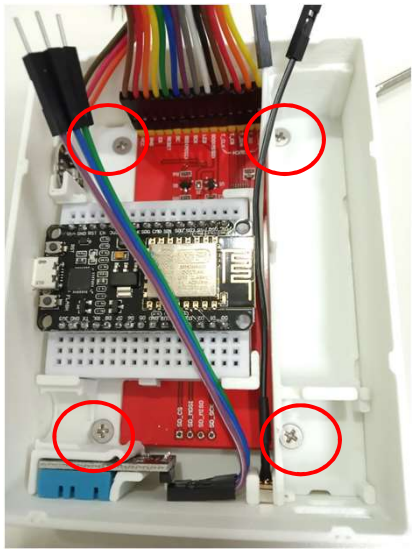


Attach Switch on the Battery Case

Place Battery Case into the case

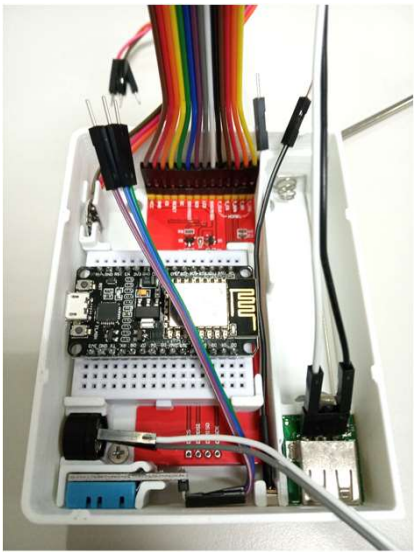
Lock 4 Screws

117



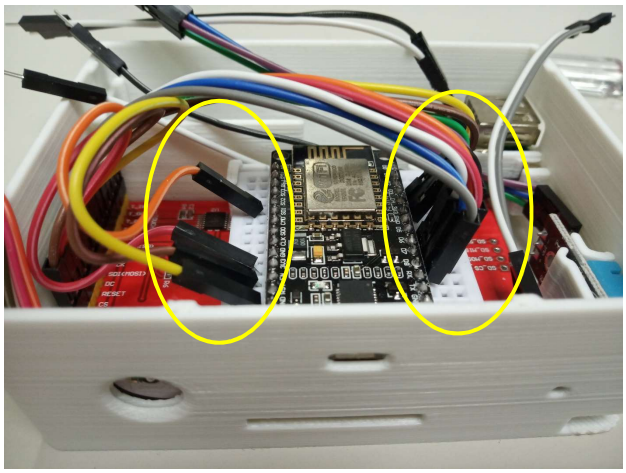
Place Speaker & Charge Module

118



Wiring

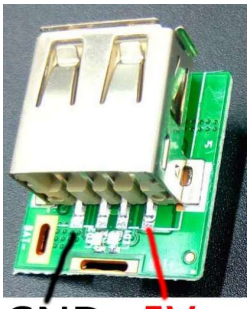
Oblique insertion



Wiring

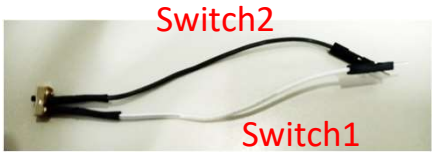
Pin on Touch screen	Pin on ESP8266
T_IRQ	D2
T_DO	D6
T_DIN	D7
T_CS	D3
T_CLK	D5
SDO(MISO)	D6
LED	D8
SCK	D5
SDI(MOSI)	D7
DC	D4
RESET	RST
CS	SD3
GND	GND
VCC	3V3

Wiring

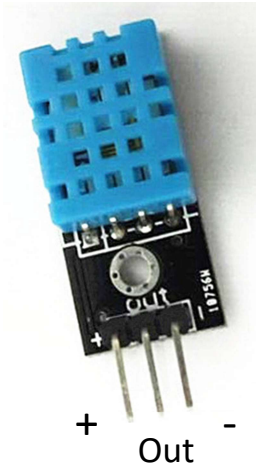


Pin on Battery Module	Pin on ESP8266
GND	GND
5V	Switch1

Pin on Switch	Pin on ESP8266
Switch1	5V on Charge Module
Switch2	Vin

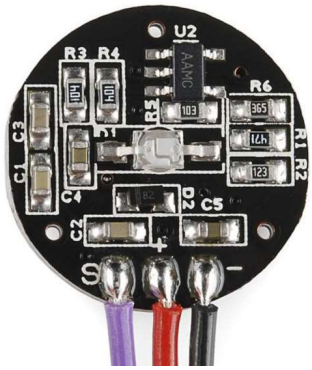


Wiring



Pin on Humidity Sensor	Pin on ESP8266
+	3V3
Out	D0
-	GND

Wiring



S + -

Pin on Heartbeat Sensor	Pin on ESP8266
S	A0
+	D8
-	GND

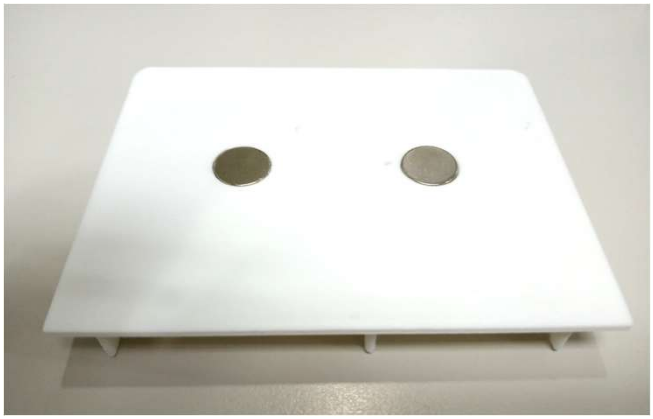
Wiring

Pin on Speaker	Pin on ESP8266
+	D1
-	GND

Wiring

Place Battery

Paste magnets onto the Back Cover



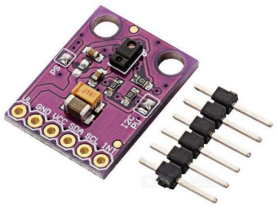
Place Back Cover and Touch Pen

Supplementary Components

Ultrasonic distance sensor

Tutorial
https://www.hackster.io/AskSensors/hc-sr04-ultrasonic-distance-with-esp8266-asksensors-iot-e4ded9

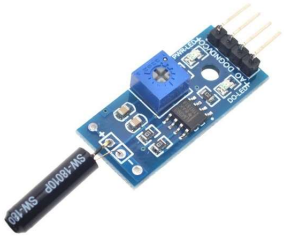
RGB and gesture sensor



Tutorial
https://learn.sparkfun.com/tutorials/apds-9960-rgb-and-gesture-sensor-hookup-guide?_ga=2.256139918.400546597.1562045640-728027900.1562045640

Supplementary Components

Vibration Sensor



Tutorial
https://www.teachmemicro.com/vibration-sensor-tutorial/

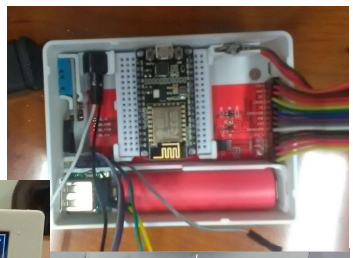
SG90 Servo Motor

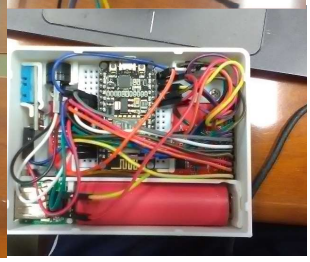
Tutorial
https://www.instructables.com/id/Interfacing-Servo-Motor-With-NodeMCU/

Arduino Basic

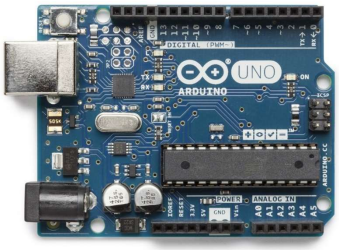
Objective

- A Game Baby
  - Micro-controller
  - Circuit
  - Sensors
  - Programs



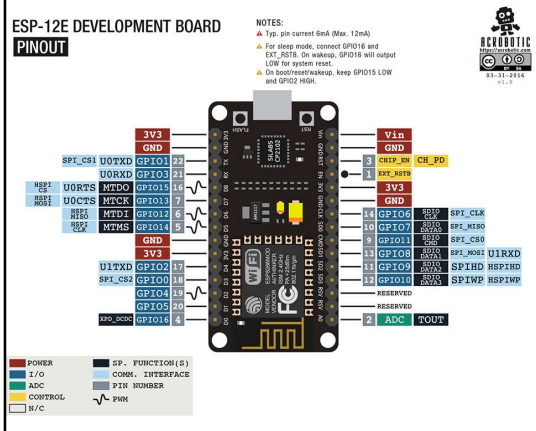


ESP8266 (WiFi) vs. UNO



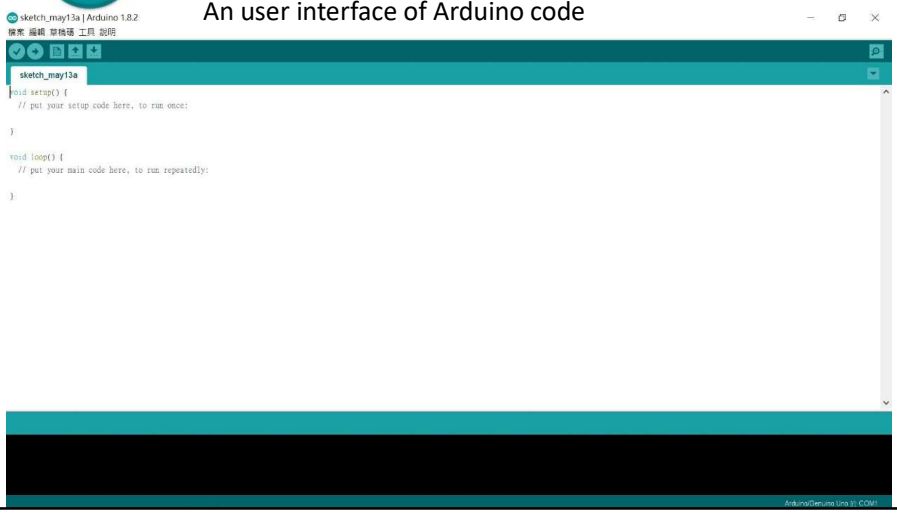
Arduino UNO
Developer: Arduino cc
Manufacturer: Many
Type: Single-board micro-controller (open source)
Operating system: none
CPU: Microchip AVR
Memory: SRAM
Storage: Flash, EEPROM

Pins for ESP8266 (WiFi) and Arduino R3

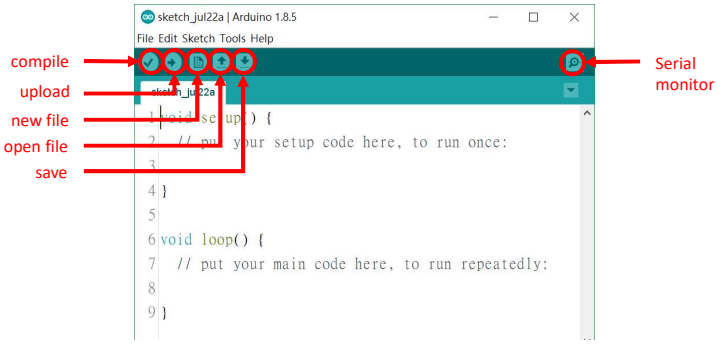


Arduino IDE

IDE: Integrated Development Environment
An user interface of Arduino code

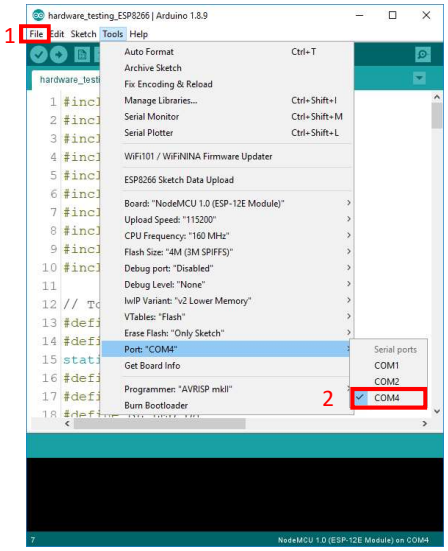


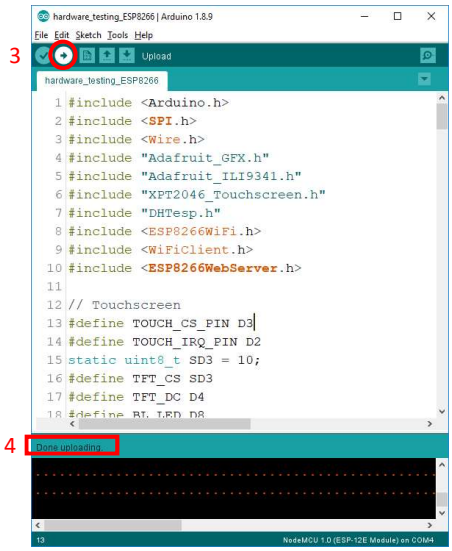
Arduino IDE



Test Your Hardware Assembly

hardware_testing_ESP8266.ino





Test Your Hardware Assembly and Touch Panel Calibration

hardware_testing_ESP8266.ino

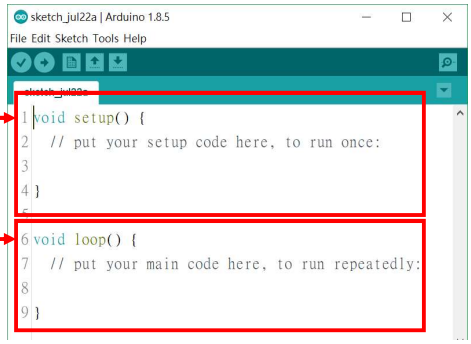
- Touch the yellow border to get

MINX, MINY, MAXX, MAXY

Arduino IDE

Run only once for program initialization

Run repeatedly



pinMode(pin,mode)

- This command must be in setup().
- Used to set Arduino pin mode (INPUT/OUTPUT).

- Ex. pinMode(D8,OUTPUT)
  → Set pin D8 as an output pin

digitalWrite(pin,value)

- Used to output voltage signal. Only HIGH/LOW for value.
- Ex. digitalWrite(D8,HIGH)
  → Output HIGH voltage (usually 5V) from pin D8

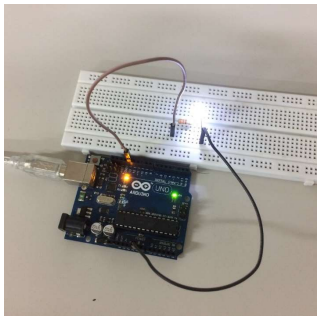
- digitalRead(pin) → Read voltage value (HIGH/LOW) from pin
- analogWrite(pin,value) → Output voltage signal. 0~255 for value
- analogRead(pin) → Read voltage value (0~255) from pin

Delay(microsecond)

- Pause the program.
- Ex. delay(1000)
  → the program will pause for 1000 milliseconds (1 second)
  (=1000*10^{-3}s)

Example

If pin D8 is connected to an LED, this program will let the LED ON for 1 second and OFF for 1 second, repeatedly.



Blink §

```
1 void setup() {
2   pinMode(D8, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(D8, HIGH);
7   delay(1000);
8   digitalWrite(D8, LOW);
9   delay(1000);
10 }
```

Basic Arduino Program statement

- Variable Declaration

int A = 5; →Declare an ***integer*** named 'A', its value is 5

double A = 5.0; →Declare an ***double-precision floating-point number*** named 'A', its value is 5.0

char A = 'a'; →Declare an ***character*** named 'A', its content is a

String A = "abc"; →Declare an ***string*** named 'A', its content is abc

Basic Arduino
Program statement

- If-else statement

```
if (condition1){
      do something;
}
else if (condition1){
      do something;
}
else if (condition1){
      do something
}
            .
            .
else{
      do something;
}
```

```
if (condition){
     do something;
}
```

```
if (condition){
      do something;
}
else{
      do something;
}
```

Basic Arduino
Program statement

- For loop statement

```
for (int i = 0, i<3, i = i + 1){
      do something;
}
```

i =	0	1	2	3
	do something	do something	do something	do nothing
				Jump up of loop

Basic Arduino
Program statement

- Function

Return type

```
int func_name(parameter1, parameter2,……){
    do something;
    return an integer;
}
```

No return

```
void func_name(parameter1, parameter2,……){
    do something;
}
```

Adafruit_GFX Display

TOUCH	Pin on NodeMCU 1.0 ESP8266
T_IRQ	D2 (GPIO4) [For Waking Up from Sleep Mode]
T_DO	D6 (GPIO12) (HMISO)
T_DIN	D7 (GPIO13) (HMOSI)
T_CS	D3 (GPIO0)
T_CLK	D5 (GPIO14) (HSCLK)

TFTLCD	Pin on NodeMCU 1.0 ESP8266
SDO (MISO)	D6 (GPIO12) (HMISO)
LED	D8 (GPIO15)
SCK	D5 (GPIO14) (HSCLK)
SDI (MOSI)	D7 (GPIO13) (HMOSI)
D/C	D4 (GPIO2)
RESET	RST
CS	SD3 (GPIO10) (SDD3)
GND	GND
VCC	3V3 (3.3V)

Graphic Display

- We use Adafruit_GFX display in this course.
- Arduino IDE doesn't have build-in commands/functions for Adafruit_GFX display.
  → we need to "include" the Adafruit_GFX library into the program
  → #include<Adafruit_GFX.h>
- Declare the object as:
  → Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

object name

pin name
Must be 10
In this course

pin name
Must be D4
In this course

tft.begin()

- Initialize the Adafruit_GFX object, must be in setup().

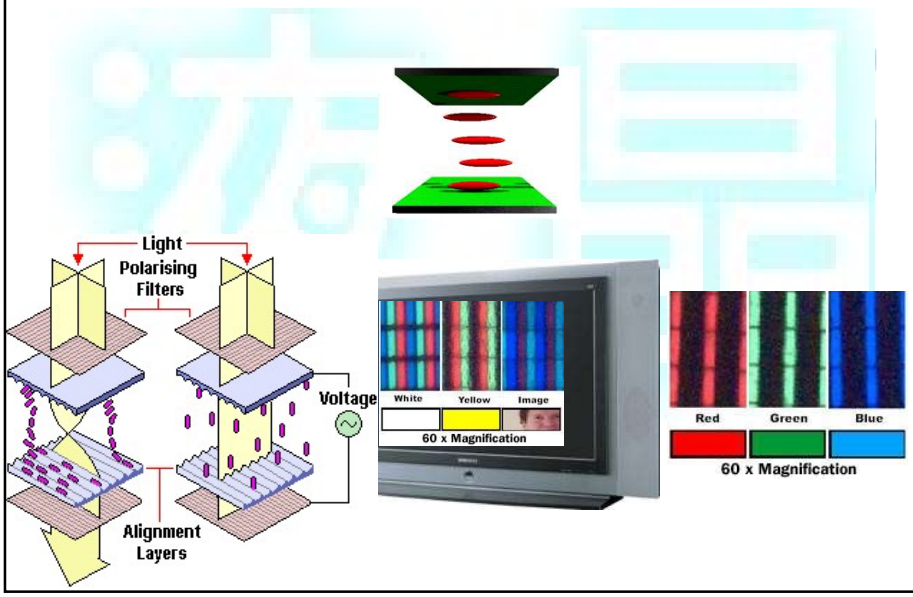
tft.fillScreen(color)

- Fill whole screen with color
- Color name:
  - ILI9341_RED
  - ILI9341_BLUE
  - ILI9341_YELLOW
  - ILI9341_BLACK
  - ILI9341_WHITE
  - ILI9341_GREEN

  etc.

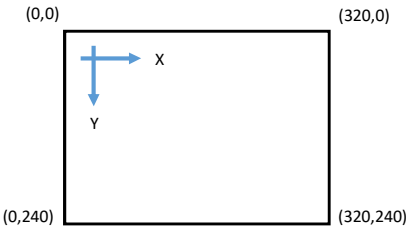
```
// Color definitions
#define ILI9341_BLACK       0x0000      /*   0,   0,   0 */
#define ILI9341_NAVY        0x000F      /*   0,   0, 128 */
#define ILI9341_DARKGREEN   0x03E0      /*   0, 128,   0 */
#define ILI9341_DARKCYAN    0x03EF      /*   0, 128, 128 */
#define ILI9341_MAROON      0x7800      /* 128,   0,   0 */
#define ILI9341_PURPLE      0x780F      /* 128,   0, 128 */
#define ILI9341_OLIVE       0x7BE0      /* 128, 128,   0 */
#define ILI9341_LIGHTGREY   0xC618      /* 192, 192, 192 */
#define ILI9341_DARKGREY    0x7BEF      /* 128, 128, 128 */
#define ILI9341_BLUE        0x001F      /*   0,   0, 255 */
#define ILI9341_GREEN       0x07E0      /*   0, 255,   0 */
#define ILI9341_CYAN        0x07FF      /*   0, 255, 255 */
#define ILI9341_RED         0xF800      /* 255,   0,   0 */
#define ILI9341_MAGENTA     0xF81F      /* 255,   0, 255 */
#define ILI9341_YELLOW      0xFFE0      /* 255, 255,   0 */
#define ILI9341_WHITE       0xFFFF      /* 255, 255, 255 */
#define ILI9341_ORANGE      0xFD20      /* 255, 165,   0 */
#define ILI9341_GREENYELLOW 0xAFE5      /* 173, 255,  47 */
#define ILI9341_PINK        0xF81F
```

Liquid Crystal Display



tft.setCursor(x,y)

- Set the position of the cursor.

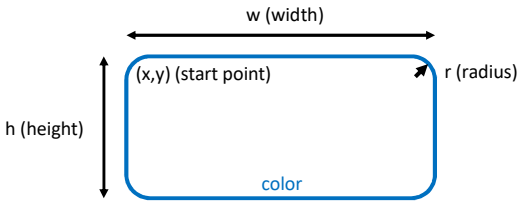


Print text on the display

- tft.print(“text”) → print “text” in the cursor position
- tft.println(“text”) → print “text” in the cursor position and set newline
- tft.setTextColor(color) → set color of text
- tft.setTextSize(size) → set size of text

tft.drawRoundRect(x,y,w,h,r,color)

- Draw a Rounded Rectangle.



tft.drawLine(x0,y0,x1,y1,color)

- Draw a line from (x0,y0) to (x1,y1).

Exercise 1-1

- graphictest.ino

```
27 void loop() {
28   /*You can modify this part*/
29   tft.setRotation(3);
30   tft.fillScreen(ILI9341_RED);
31   delay(1000);
32   tft.fillScreen(ILI9341_BLACK);
33   tft.setCursor(50,50);
34   tft.setTextColor(ILI9341_WHITE);
35   tft.setTextSize(2);
36   tft.println("Hello World!");
37   delay(1000);
38   /*You can modify this part*/
39 }
```

Touchscreen

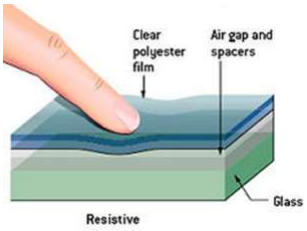
- Use XPT2046_Touchscreen in this course.
- Need to #include< XPT2046_Touchscreen .h> in the program.
- Declaration as:

  XPT2046_Touchscreen ts(TOUCH_CS_PIN,TOUCH_IRQ_PIN);

- Resistive touch screen

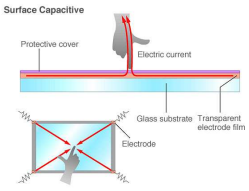
Touchscreen

- Resistive touchscreen:
  - When someone presses the touch panel, the top layer bends to make contact with the bottom layer, closing a circuit and causing a current loop
  - Single-touch device.
  - Less cost.
  - Simple.
  - Low touch sensitivity.



Touchscreen

- Capacitive touchscreen:
  - Distinguish and sense specific touch location based on the electrical impulses in a human body, by measuring the resulting changes in electrostatic capacity.
  - Support multi-touch.
  - More Expensive.
  - High touch sensitivity.



ts.begin()

- Initialize the touchscreen object, must be in setup().

ts.touched()

- Determine whether the panel been touched.
- Ex.

```
if(ts.touched())
{
      /*do something*/
}
else
{
      /*do something*/
}
```

if-else statement:
- Run program with some condition
- if(condition)
  { /*do something*/ }
  else
  { /*do something*/ }

ts.getPoint()

- Get the position been touched.
- This command return a (x,y) pair
- Ex. TS_Point p = ts.getPoint();

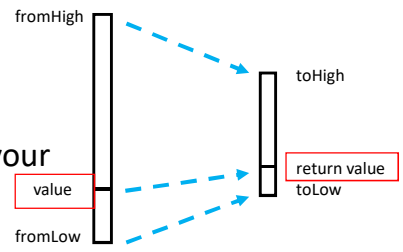
- The coordinate of the touch panel is different from the display!!!
  → use map() to convert the touch panel coordinate to the display coordinate.

map(value,fromLow,fromHIGH,toLow,toHIGH)

- Map a value in interval [fromLow,fromHigh] to interval [toLow,toHIGH]
- Linearly.
- Ex.

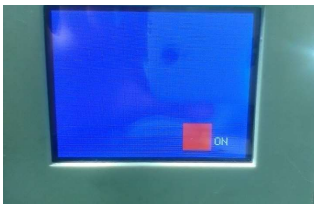
```
p.y = map(p.y, TS_MINY, TS_MAXY, 0, tft.height());
p.x = map(p.x, TS_MINX, TS_MAXX, 0, tft.width());
```

- Use calibration program to get your
  - TS_MINX, TS_MAXX
  - TS_MINY, TS_MAXY



Exercise 1-2 On/Off Buttons

- onoffbutton.ino



Exercise 1-2 On/Off Buttons

- onoffbutton.ino

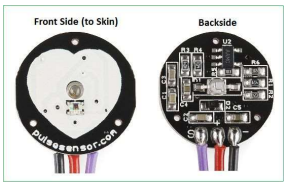
Determine whether it is touched and where is being touched.

Do something based on the touched point

```
98  if (ts.touched() && !beentouched)
99  {
100   beentouched = true;
101   // Retrieve a point
102   TS_Point p = ts.getPoint();
103
104   p.y = map(p.y, TS_MINY, TS_MAXY, 0, tft.height());
105   p.x = map(p.x, TS_MINX, TS_MAXX, 0, tft.width());
106   int y = p.y;
107   int x = p.x;
109 if (RecordOn)
110 {
111   if((x > REDBUTTON_X) && (x < (REDBUTTON_X + REDBUTTON_W))) {
112     if ((y > REDBUTTON_Y) && (y <= (REDBUTTON_Y + REDBUTTON_H))) {
113       Serial.println("Red btn hit");
114       redBtn();
115     }
116   }
117 }
118 else //Record is off (RecordOn == false)
119 {
120   if((x > GREENBUTTON_X) && (x < (GREENBUTTON_X + GREENBUTTON_W))) {
121     if ((y > GREENBUTTON_Y) && (y <= (GREENBUTTON_Y + GREENBUTTON_H))) {
122       Serial.println("Green btn hit");
123       greenBtn();
124     }
125   }
```

Heartbeat sensor

- Use green light LED and a light sensor to measure the heart pulse.
- Hemoglobin (the protein that makes the blood red) will absorb the green light.
- Measure the reflection rate of the green light from the fingertip.
- The period of the reflection rate is approximately the heart pulse rate.



analogRead(PIN)

- Use this command to get analog voltage value at PIN.
- In this course, we use:

  analogRead(A0);

  to get voltage at A0, which is connected to the output of heartbeat sensor.

tone(PIN,freq,duration) & noTone()

- Use tone(PIN,freq,duration) to generate beep.
- Use noTone() to stop beep.
- Frequency(Hz) only support integer.
- Duration(ms) is optional.
- Ex. tone(D1,392)

tone(PIN,freq,duration) & noTone()

Note	Great	Small	One-lined	Two-lined	Three-lined	Four-lined
A	55.00	110.00	220.00	440.00	880.00	1760.00
A♯/B♭	58.27	116.54	233.08	466.16	932.33	1864.66
B/C♭	61.74	123.47	246.94	493.88	987.77	1975.53
B♯/C	65.41	130.81	261.63	523.25	1046.50	2093.00
C♯/D♭	69.30	138.59	277.18	554.37	1108.73	2217.46
D	73.42	146.83	293.66	587.33	1174.66	2349.32
D♯/E♭	77.78	155.56	311.13	622.25	1244.51	2489.02
E/F♭	82.41	164.81	329.63	659.26	1318.51	2637.02
E♯/F	87.31	174.61	349.23	698.46	1396.91	2793.83
F♯/G♭	92.50	185.00	369.99	739.99	1479.98	2959.96
G	98.00	196.00	392.00	783.99	1567.99	3135.96
G♯/A♭	103.83	207.65	415.30	830.61	1661.22	3322.44

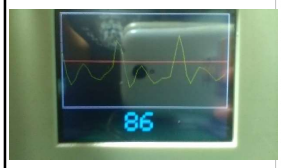
Exercise 1-3 Heartbeat

- ILI9341_Heartbeat_ESP8266.ino

analogRead

```
54 int getValue(void) {
55   int value = -1;
56   value = analogRead(PulseSensorPin);
57   return value;
58 }
```

Calculate heartbeat frequency. Check whether the voltage value is higher than a threshold.



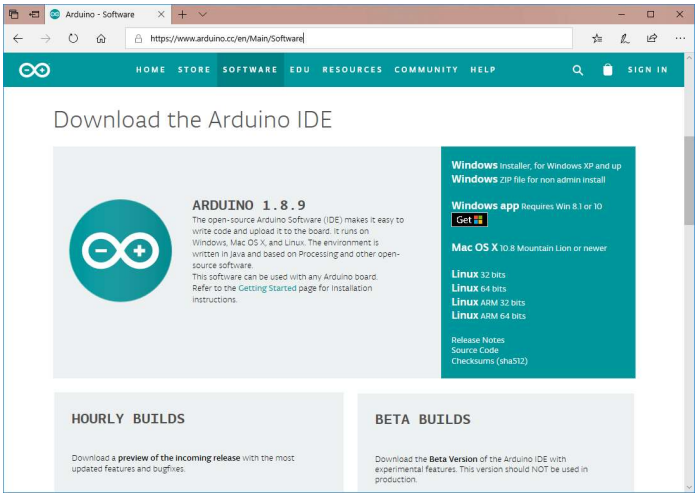

```
109   nowtime = millis();
110   if ((values[DATA_POINTS-1] > (threshold+deadzone))&&(values[DATA_POINTS-2]<=(threshold-deadzone))
111   {
112     timeStamp[stableCount] = nowtime;
113     tft.setCursor(80, 200);
114     tft.setTextColor(ILI9341_RED);
115     tft.setTextSize(4);
116     tft.print("o");
117     tone(SPK_PIN,3520);
118     pulseRate = (int)(60000.0 /(float)(nowtime-prevtime));
119     if (pulseRate<220)
120     {
121       tft.fillRect(120,190,200,240,ILI9341_BLACK);
122       tft.setCursor(120, 190);
123       tft.setTextColor(ILI9341_CYAN);
124       tft.setTextSize(5);
125       tft.print(pulseRate);
126     }
127     prevtime = nowtime;
128   }
```

Appendix A

How to install Arduino IDE and libraries

Download Arduino IDE

https://www.arduino.cc/en/Main/Software

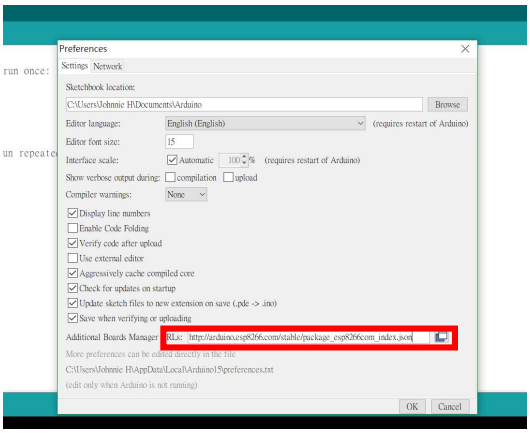


ESP8266 Setup

File -> Preference

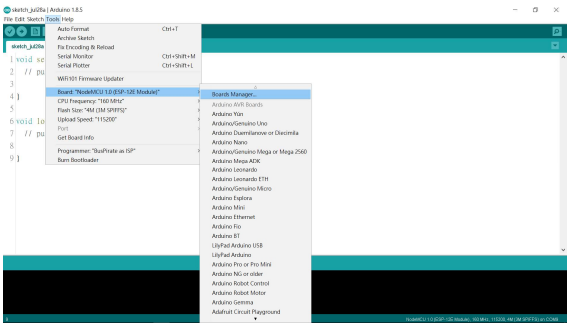
ESP8266 Setup

Additional Boards Manager URLs -> http://arduino.esp8266.com/stable/package_esp8266com_index.json



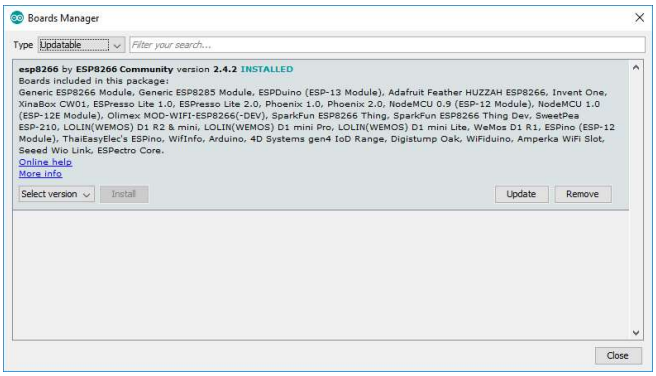
ESP8266 Setup

Tools -> Board -> Board Manager



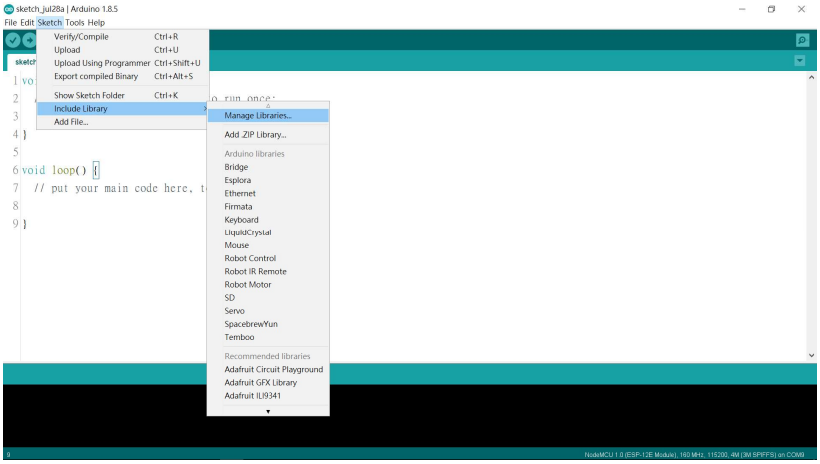
ESP8266 Setup

Search ESP8266 (ver2.4.2) and install



Libraries Setup

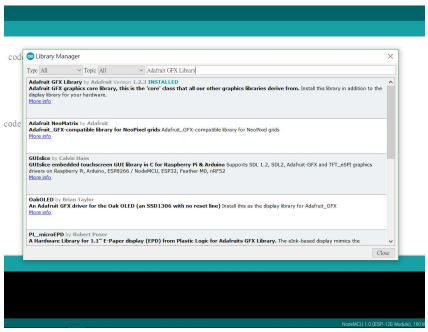
Sketch -> Include Library -> Manage Libraries…



Libraries Setup

Search and install:

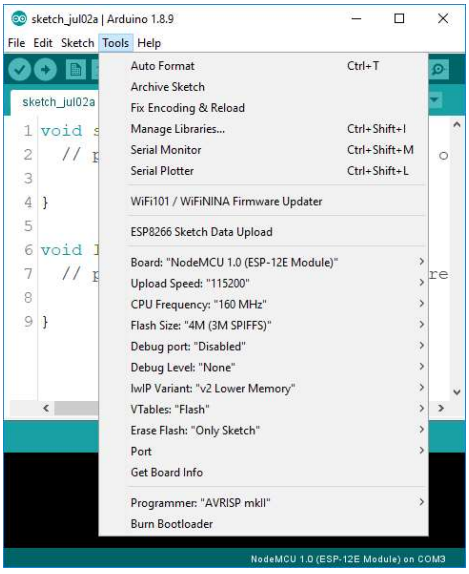
Adafruit GFX Library
Adafruit ILI9341
DHT sensor library for ESPx (beegee_tokyo ver.)
ESP8266 Weather Station (ThingPulse ver.)
Json Streaming Parser (Daniel Eichhorn ver.)
WiFiManager (tzapu ver.)
XPT2046_Touchscreen



Board Setup

Tools ->

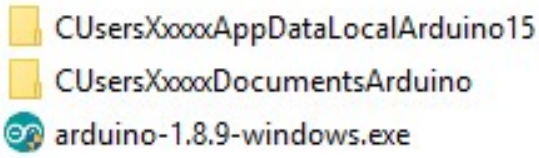
Board:	NodeMCU 1.0
Upload Speed:	115200
CPU Frequency:	160MHz
Flash Size:	4M (3M SPIFFS)



Appendix B
Quick Install

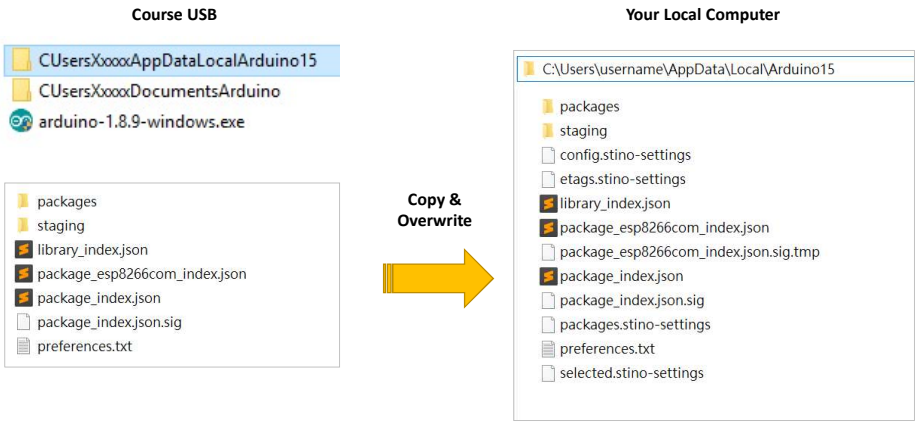

Execute Arduino Install File


USBDrive:\QuickInstallation



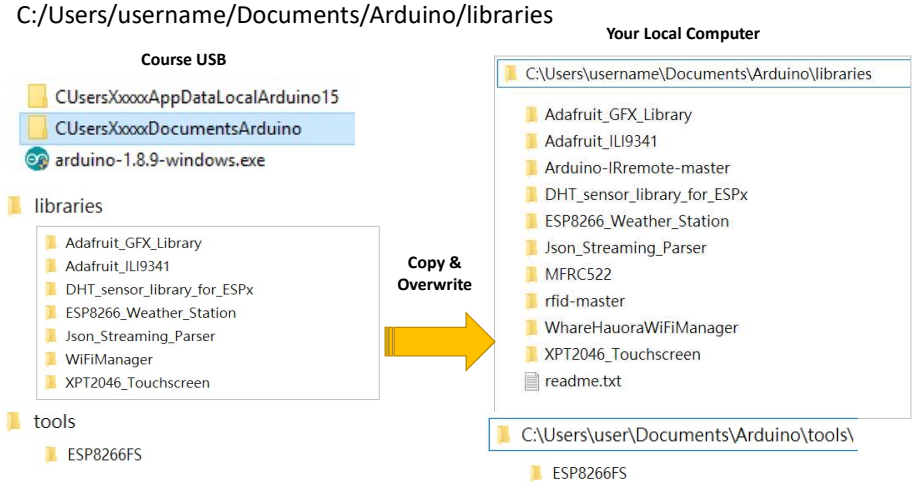
Copy file to Arduino15 folder

Open “CUsersXxxxxAppDataLocalArduino15” folder
and copy all files into
C:/Users/username/Appdata/Local/Arduino15



Copy file to libraries folder

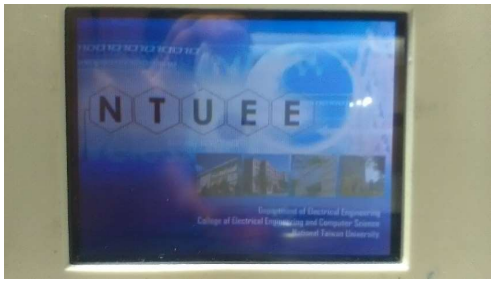
Open “CUsersXxxxxDoucumentsArduinoLibraries” folder
and copy all files into
C:/Users/username/Documents/Arduino/libraries



Appendix C
How to show image on the display?

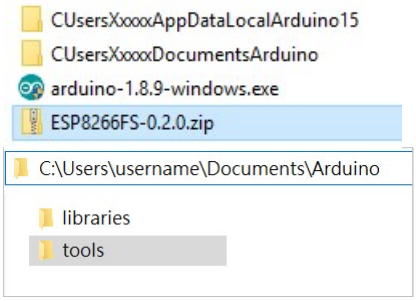
Example

- welcome.ino



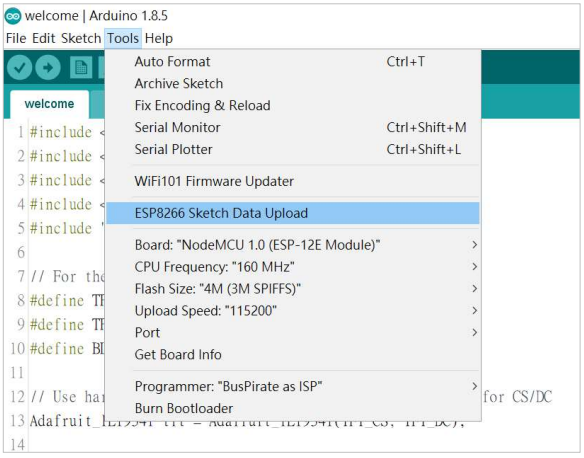
Install ESP8266FS

- https://github.com/esp8266/arduino-esp8266fs-plugin/releases/tag/0.2.0
- Download the file and unzip.
- Create “tools” folder in C:/User/username/Documents/Arduino/
- Put the unzip file into /tools



Install ESP8266FS

- Restart Arduino IDE
- You will see the new menu item **'ESP8266 Sketch Data Upload'**.



Library settings

- Copy GfxUi.cpp & GfxUi.h to your sketch directory from the weather station course directory.
- #include<SPI.h>, #include<Wire.h>, include "GfxUi.h“ in your code.
- Declare GfxUi object.
- Initialize SPIFFS object.


```
18 void setup() {
19   Serial.begin(9600);
20   Serial.println("ILI9341 Test!");
21
22   tft.begin();
23   pinMode(BL_LED,OUTPUT);
24   digitalWrite(BL_LED,HIGH);
25
26   SPIFFS.begin();
```

```
1 #include <SPI.h>
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_ILI9341.h>
5 #include "GfxUi.h"
```

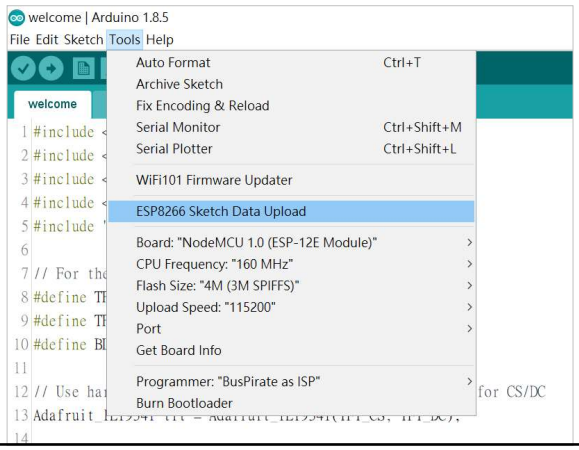
```
12 //Graphic display define
13 Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);
14
15 //UI define
16 GfxUi ui = GfxUi(&tft);
```

Upload Image to EPS8266

- Create a sub-directory within your sketch directory named 'data‘.
- Put images into /data.
- The image size must be smaller than 320x240.
- Use .bmp format.

Upload Image to EPS8266

- Connect ESP8266 to computer, press **'ESP8266 Sketch Data Upload'**.
- Wait for upload process.

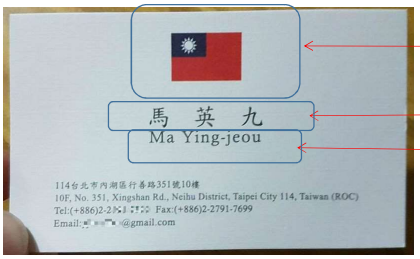


Display Image on EPS8266

- ui.drawBmp("filename",x,y)
- This command will display the image "filename" at (x,y).

```
35 void loop() {
36   /*tft.drawRoundRect(0,0,320,240,10,ILI9341_WHITE);
37   tft.setCursor(130,90);
38   tft.setTextSize(2);
39   tft.print("NTUEE");*/
40   tft.setCursor(10,200);
41   tft.setTextSize(2);
42   tft.print("Welcome!!");
43   yield();
44   ui.drawBmp("/welcome.bmp",0,0);
45   yield();
46 }
```

Bonus Exercise 1
Name Card



Your photo “BMP”

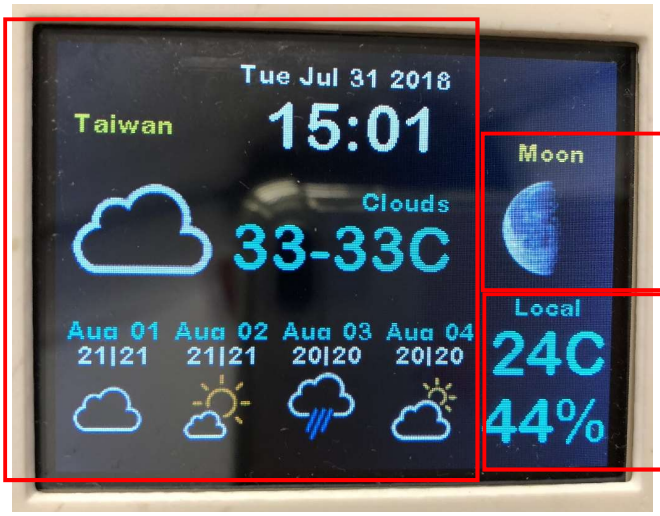
Your name in Arabian or Chinese

Your English name “text”

Weather Station

• Home Weather Station

OpenWeatherMap

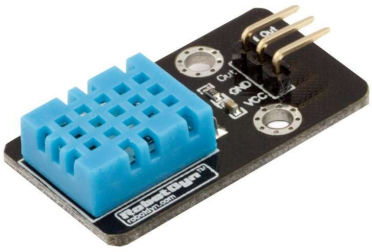


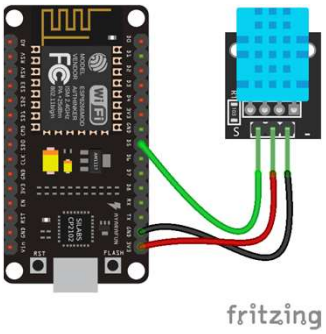
Moon Phase

DHT

• DHT

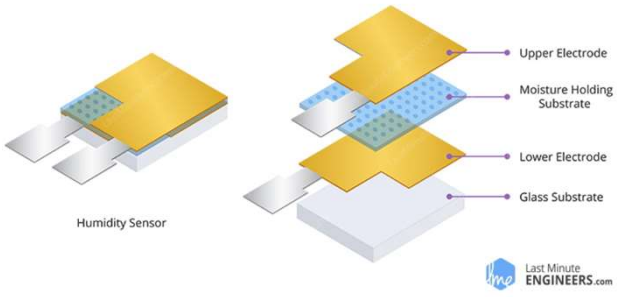
The DHT11 is a temperature and humidity sensor.





• DHT11

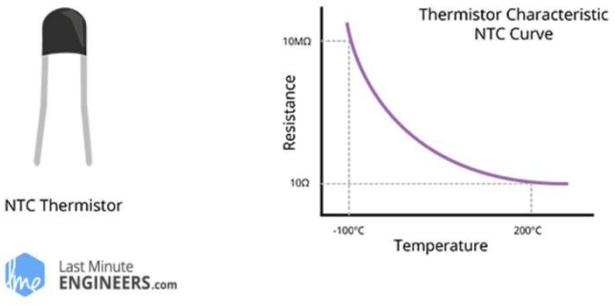
- Humidity sensing component
  1.The ions are released by the substrate as water vapor is absorbed by it
  ->increases the conductivity between the electrodes.
  2. The change in resistance between the two electrodes is proportional to the relative humidity. (Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.)



Internal Structure of Humidity Sensor

• DHT11

- NTC temperature sensor/Thermistor
  1.A thermistor is a thermal resistor.Comparing with general resistors, it changes drastically with temperature.
  2.NTC:Negative Temperature Coefficient



NTC Thermistor with Characteristic Curve

• DHT

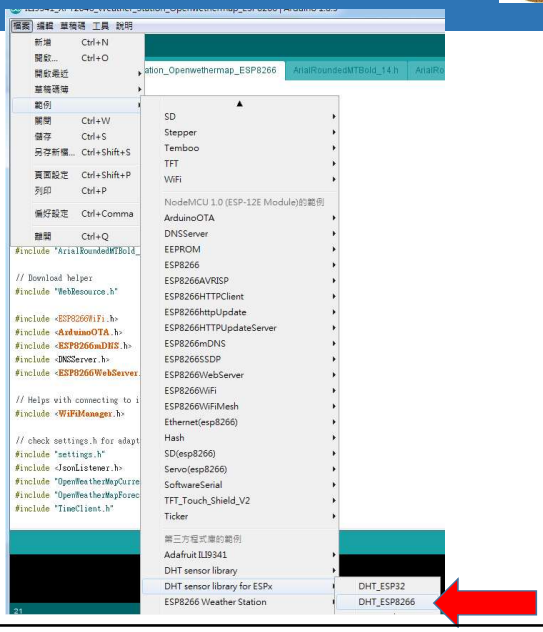
- ➢ Humidity Range: 20-80% RH
- ➢ Humidity Accuracy: ±5% RH
- ➢ Temperature Range: 0-50 °C
- ➢ Temperature Accuracy: ±2% °C
- ➢ Operating Voltage: 3V to 5.5V

• DHT

basic code

```
#include <DHTesp.h>
#define DHTPIN      D0  // what pin we're connected to
#define DHTTYPE     DHT11
  void setup() {
        dht.setup(DHTPIN, DHTesp::DHTTYPE); // Connect DHT sensor
  }
  float h = dht.getHumidity();
  float t = dht.getTemperature();
```

• DHT

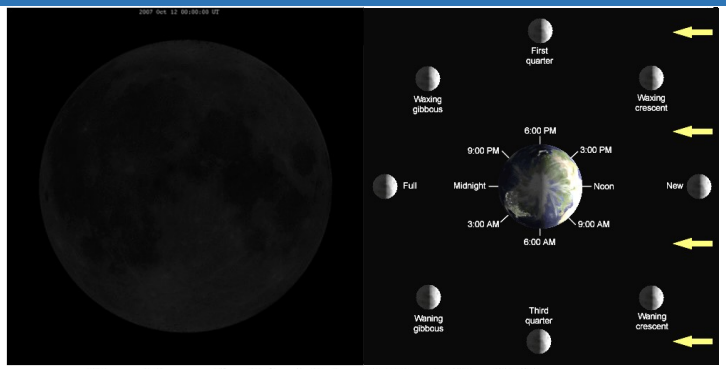


• DHT


```
dht.setup(17, DHTesp::DHT22); // Connect DHT sensor to GPIO 17
```

```
dht.setup(DO, DHTesp::DHT11); // Connect DHT sensor to  DO
```

• Moon phase



Moon Phases for Taipei, 6 Jan 2019 – 28 Jan 2019

New Moon	First Quarter	Full Moon	Third Quarter
6 January 09:28	14 January 14:45	21 January 13:16 Super Moon Wolf Moon	28 January 05:10

• Moon phase

- The lunar phase or phase of the Moon is the shape of the directly sunlit portion of the Moon as viewed from Earth.
- The lunar phases gradually and cyclically change over the period of a synodic month (about **29.53 days**), as the orbital positions of the Moon around Earth and of Earth around the Sun shift.
- New moon was shown @ 09:56 PM, Jan. 01, 1900.

• Julian Day

➢ Julian day is the continuous count of days since the beginning of the Julian Period and is used primarily by astronomers.

➢ The Julian date (JD) of any instant is the Julian day number plus the fraction of a day since the preceding noon in Universal Time.

D stands for Julian Date. 0h is 00:00 midnight, 12h is 12:00 noon, UT unless otherwise specified. Current value is as of 15:09, Saturday, January 12, 2019 (UTC) and may be cached. (update)

Name	Epoch	Calculation	Current value	Notes
Julian Date	12h Jan 1, 4713 BC		**2458496.13125**	
Reduced JD	12h Nov 16, 1858	JD − 2400000	**58496.13125**	[11][12]
Modified JD	0h Nov 17, 1858	JD − 2400000.5	**58495.63125**	Introduced by SAO in 1957
Truncated JD	0h May 24, 1968	floor (JD − 2440000.5)	**18495**	Introduced by NASA in 1979
Dublin JD	12h Dec 31, 1899	JD − 2415020	**43476.13125**	Introduced by the IAU in 1955

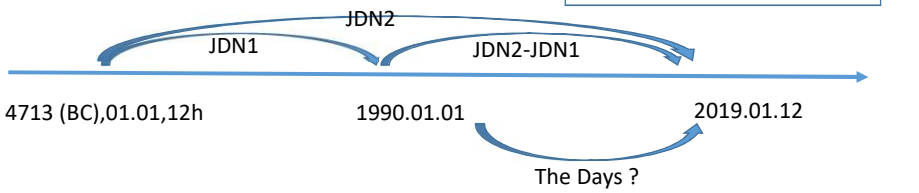
Julian Day Number

➢ The algorithm is valid for all (possibly proleptic) Gregorian calendar dates after Nov. 23, 4713 (B.C.)

➢ JDN (Y, M, D)= (1461.0 * (Y + 4800.0 + (M - 14.0)/12.0))/4.0
+(367.0 * (M - 2.0 - 12.0 * ((M - 14.0)/12.0)))/12.0
- (3.0 * ((Y + 4900.0 + (M - 14.0)/12.0)/100.0))/4.0
+ D - 32075.0

• Moon phase

Years	months	days	
2018	8	7	= 2458333.5 Julian Days JDN2
- 1900	1	1	= 2415020.5 Julian Days JDN1
118	7	6	= the Julian days since 1900/1/1 JDN2 -JDN1
		?	= the days since 1900/1/1



4713 (BC),01.01,12h 1990.01.01 2019.01.12

The Days ?

• Moon phase

(JDN2-JDN1)

118years 7months 6days ÷ 29.53days = 1467.46 periods JDN_months

1467.46 - 1467.00 = 0.46 periods

0.46 * 23.9 = 10.994

Round 10.994 to the nearest tens.

• Moon phase

Use this integer to match the file name.

ui.drawBmp("filename.bmp", X, Y)

• Moon phase


```
float JDN(int Y, int M, int D)
{ float jdn = (1461.0 * (Y + 4800.0 + (M - 14.0)/12.0))/4.0 +(367.0 * (M - 2.0 - 12.0 * ((M -
14.0)/12.0)))/12.0 - (3.0 * ((Y + 4900.0 + (M - 14.0)/12.0)/100.0))/4.0 + D - 32075.0;
  Serial.println(jdn);
  return jdn;
/* subtract integer part to leave fractional part of original jd */
 }
```

```
  float jdn_months = (JDN(timeinfo->tm_year+1900, timeinfo->tm_mon+1, timeinfo->tm_mday)-JDN(1900, 1, 1)-2)/29.53;
  float moonphase_now = jdn_months - floor(jdn_months);
  int moonAgeImage = (int) (23.9 * moonphase_now);
  ui.drawBmp("/moon" + String(moonAgeImage) + ".bmp", 250, 68);
}
```

Float (floating point)

- Floating-point arithmetic (FP) is arithmetic using formulaic representation of real numbers as an approximation so as to support a trade-off between range and precision.
- For this reason, floating-point computation is often found in systems which include very small and very large real numbers, which require fast processing times.

Float

A number that can be represented exactly is of the following form:

$$\textbf{significand} \times \textbf{base}^{\textbf{exponent}}$$

where **significand** is an integer (i.e., in Z),
base is an integer greater than or equal to two,
exponent is also an integer.

Float

➢ For example:

$$1.2345 = 12345 \times 10^{-4}$$

$10110101.1101_2 = 181.8125_{10}$

1	0	1	1	0	1	0	1	.	1	1	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}
2^7	+	2^5	+	2^4	+	2^2	+	2^0	+	2^{-1}	+	2^{-2}	+	2^{-4}
=	128	+	32	+	16	+	4	+	1	+	0.5	+	0.25	+	0.0625
=	181.8125														

Floor

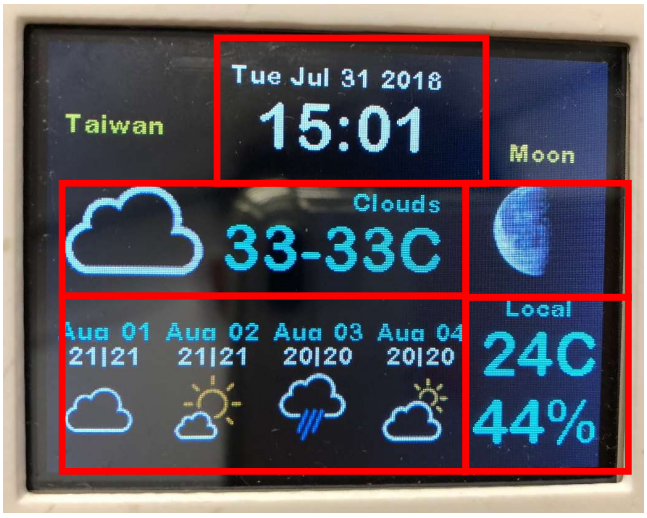
- The floor function is the function that takes as input a real number x and gives as output the greatest integer less than or equal to x

Floor

- For example,

  floor(2.4) = [2.4] = 2

• Layout



drawString

➢ You can simply use drawString() to display a message at a given set of coordinates in a given colour:

ex.

ui.drawstring(X ,Y ,Text)

drawString

➢ Example:
ui.drawstring(X ,Y ,Text)

x : x-axis of starting position on the screen.
y : y-axis of starting position on the screen.
Text is the string to be displayed.

·drawBMP

➢ Display a bmp file on the LCD screen. The bmp file should be stored in sd card. BMP format should be 24-bit color, and no bigger than 128*160. If an image dimensions exceed the size of the screen, it will be cropped.

drawBMP

➢ Example:

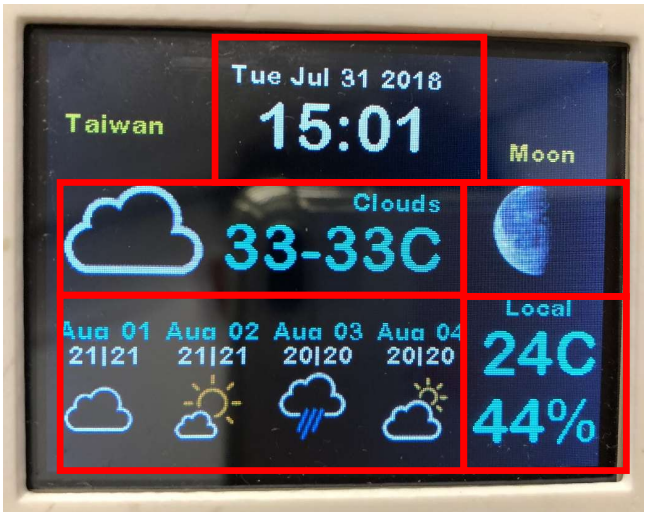
ui.drawBMP(filename , x , y)

filename : name of the bmp to be displayed.

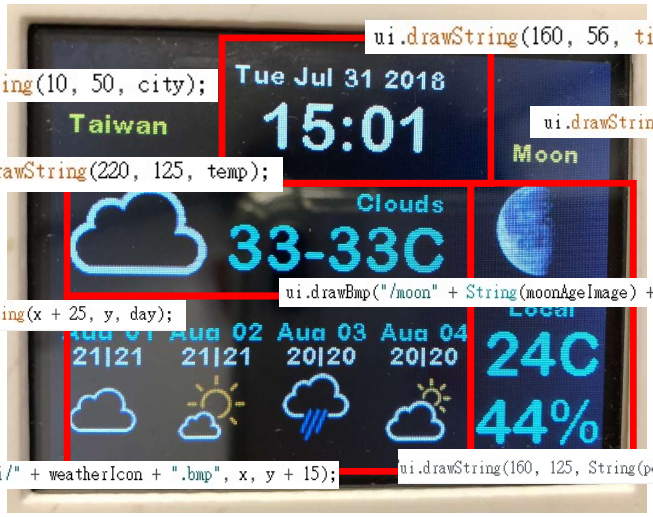
x : x-axis of starting position on the screen.

y : y-axis of starting position on the screen.

• Layout



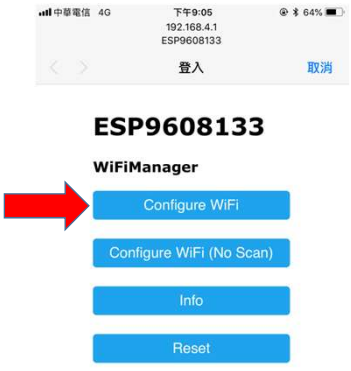
• Layout



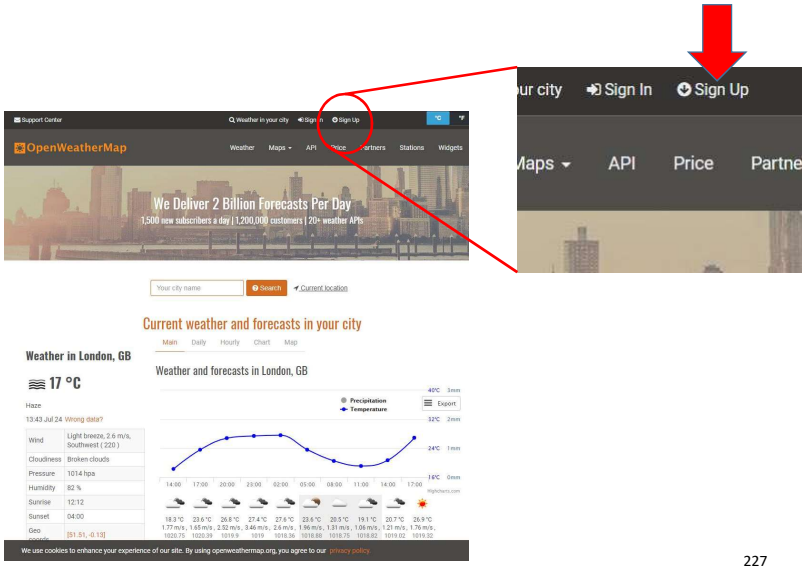
• WiFi Setting

- WiFi Setting

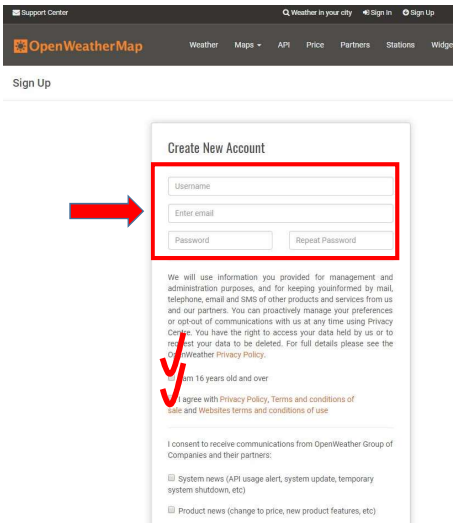
- WiFi Setting



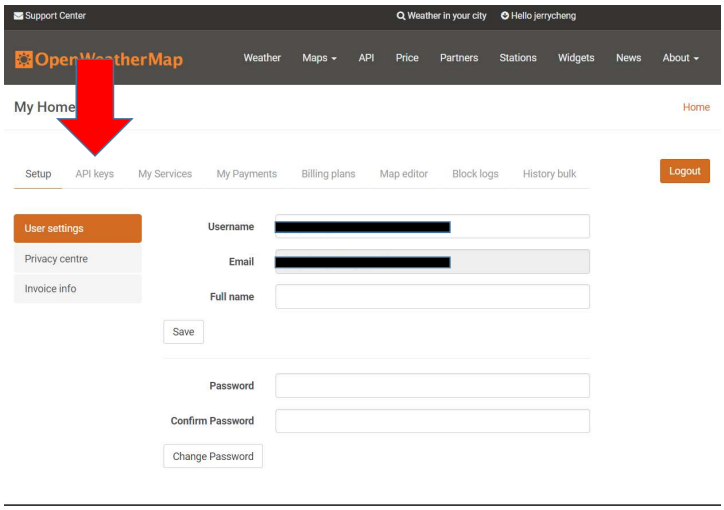
• OpenWeatherMap



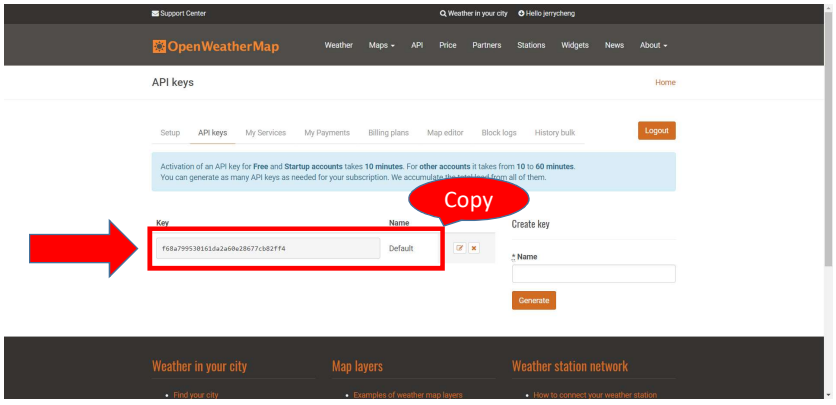
• OpenWeatherMap



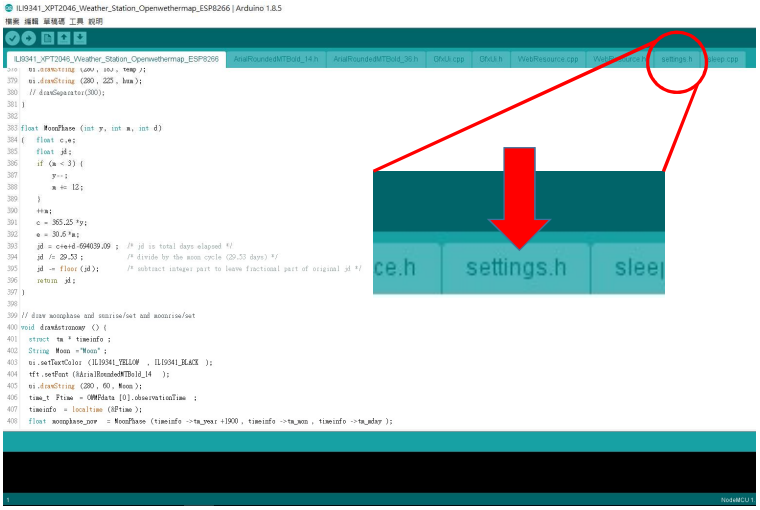
• OpenWeatherMap



• OpenWeatherMap



• OpenWeatherMap

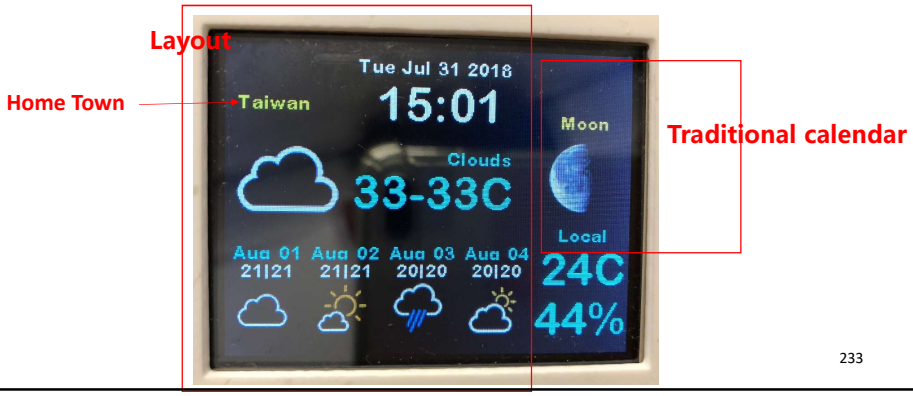


• OpenWeatherMap

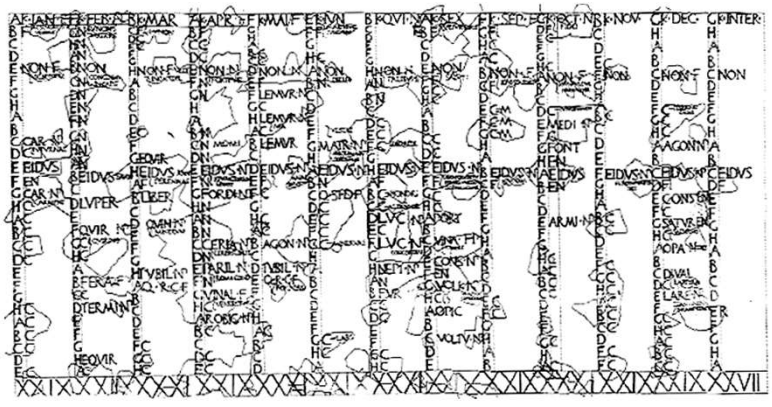

```
// Setup
const int UPDATE_INTERVAL_SECS = 10 * 60;  // Update every 10 minutes
boolean USE_TOUCHSCREEN_WAKE = true;       // use the touchscreen to wake up, ~90mA current draw
boolean DEEP_SLEEP = false;                  // use the touchscreen for deep sleep, ~10mA current draw but doesnt work
int     AWAKE_TIME = 5;                    // how many seconds to stay 'awake' before going back to zzz

// Pins for the ILI9341
static uint8_t SD3 = 10;
#define TFT_CS SD3
#define TFT_DC D4
#define BL_LED D8

// TimeClient settings
const float UTC_OFFSET = 8;

const boolean IS_METRIC = true;

// Openweathermap Settings
const String OPENWEATHERMAP_APP_ID = "Your_API_key";
const String OPENWEATHERMAP_LANGUAGE = "EN";
const String OPENWEATHERMAP_COUNTRY = "TW";
const String OPENWEATHERMAP_CITY = "Taiwan";

// Thingspeak Settings
const String THINGSPEAK_CHANNEL_ID = "67284";
const String THINGSPEAK_API_READ_KEY = "L2VIW20QVWZJBLAK";
```

• Bonus Exercise 2

By modifying the code we just learned,
please change
(1) the layout of the weather station
(2) the weather information of your home town
(3) your traditional calendar.



• Roman Calendar



Roman Calendar

- The Roman calendar was the calendar used by the Roman kingdom and republic.
- Original calendar consisted of 10 months beginning in spring with March; winter was left as an unassigned span of days.
- The winter period was later divided into two months , January and February.

- Islamic Calendar

1- Muharram

9- September			10- October			
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 11	2 12	3 13	4 14	5 15
6 16	7 17	8 18	9 19	10 20	11 21	12 22
13 23	14 24	15 25	16 26	17 27	18 28	19 29
20 30	21 1	22 2	23 3	24 4	25 5	26 6
27 7	28 8	29 9				

2- Safar

10- October			11- November			
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1 10	2 11	3 12	4 13
5 14	6 15	7 16	8 17	9 18	10 19	11 20
12 21	13 22	14 23	15 24	16 25	17 26	18 27
19 28	20 29	21 30	22 31	23 1	24 2	25 3
26 4	27 5	28 6	29 7	30 8		

3- Rabi I

11- November			12- December			
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1 9	2 10
3 11	4 12	5 13	6 14	7 15	8 16	9 17
10 18	11 19	12 20	13 21	14 22	15 23	16 24
17 25	18 26	19 27	20 28	21 29	22 30	23 1
24 2	25 3	26 4	27 5	28 6	29 7	

4- Rabi II

12- December			1- January			
Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1 8
2 9	3 10	4 11	5 12	6 13	7 14	8 15
9 16	10 17	11 18	12 19	13 20	14 21	15 22
16 23	17 24	18 25	19 26	20 27	21 28	22 29
23 30	24 31	25 1	26 2	27 3	28 4	29 5
30 6						

5- Jumada I

1- January			2- February			
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1 7	2 8	3 9	4 10	5 11	6 12
7 13	8 14	9 15	10 16	11 17	12 18	13 19
14 20	15 21	16 22	17 23	18 24	19 25	20 26
21 27	22 28	23 29	24 30	25 31	26 1	27 2
28 3	29 4	30 5				

6- Jumada II

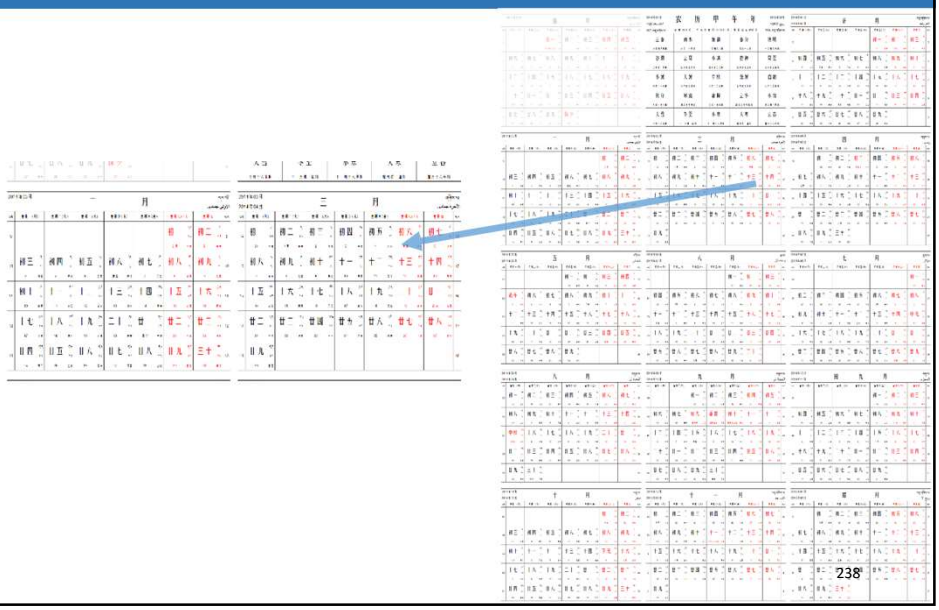
2- February			3- March			
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1 6	2 7	3 8	4 9
5 10	6 11	7 12	8 13	9 14	10 15	11 16
12 17	13 18	14 19	15 20	16 21	17 22	18 23
19 24	20 25	21 26	22 27	23 28	24 1	25 2
26 3	27 4	28 5	29 6	30 7		

Islamic Calendar

➢ The Islamic, Muslim, or Hijri calendar is a lunar calendar consisting of 12 months in a year of 354 or 355 days.

➢ The first day of the first month of the Islamic calendar (1 Muharram 1 AH) was set to the first new moon after the day the Prophet moved from Quba' to Medina (originally 26 Rabi' I on the pre-Islamic calendar) i.e., Friday, 16 July AD 622, the equivalent civil tabular date (same daylight period) in the Julian calendar.

• Chinese Calendar



Chinese Calendar

- The traditional Chinese calendar (official Chinese name: Rural Calendar, alternately Former Calendar , Traditional Calendar , or Lunar Calendar) is a lunisolar calendar which reckons years, months and days according to astronomical phenomena.

- For each month, there would be characters for if the month was long (containing 30 days) or short (containing 29 days); the stem-branches for the first, eleventh, and 21st days; and the date, stem-branch, and time of the solar terms.

Arduino Calculator

Simple Calculator

- Include +-*/ operator.
- Calculate 2 integers only.
- Real time display.

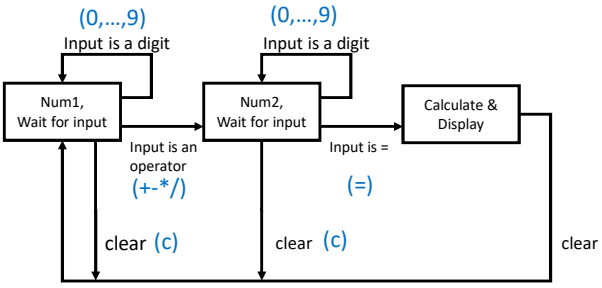
Input

- Digits (0~9)
- Operators (+-*/)
- Equal (=)
- Clear (C)

Input parsing strategy

Man Work

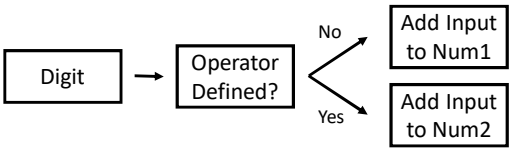
Num1 + Num2 =
23 + 15 =



Input parsing strategy

Computer Work

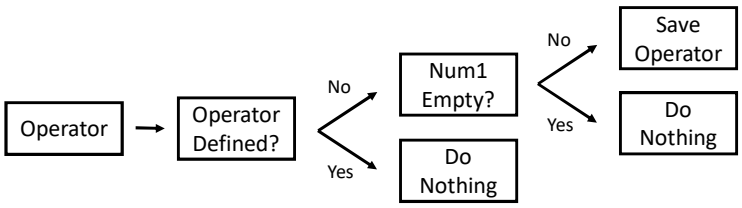
Num1 + Num2 =



Input parsing strategy

Computer Work

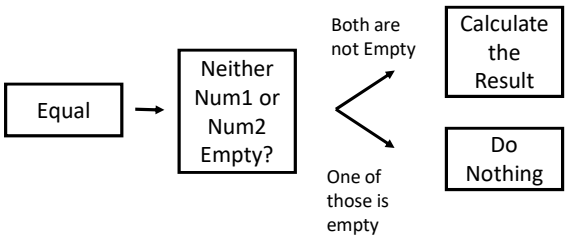
Num1 + Num2 =



Input parsing strategy

Computer Work

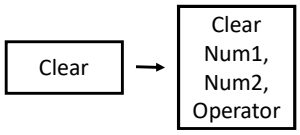
Num1 + Num2 =



Input parsing strategy

Computer Work

Num1 + Num2 =



Input parsing strategy

Computer Work

Example: 23+15=

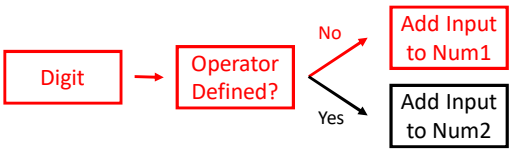
Num1:
Num2:
Operator:

Input parsing strategy

Computer Work

Example: 23+15=

Num1: 2
Num2:
Operator:

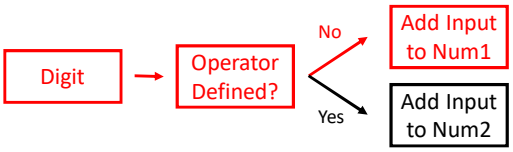


Input parsing strategy

Computer Work

Example: 23+15=

Num1: 23
Num2:
Operator:

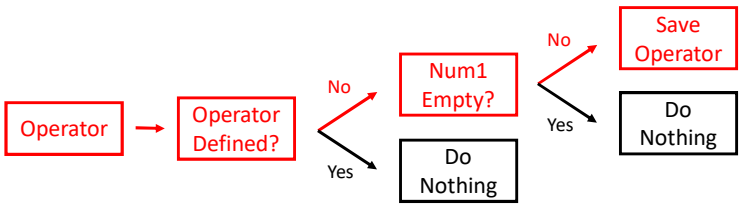


Input parsing strategy

Computer Work

Example: 23+15=

Num1: 23
Num2:
Operator: +

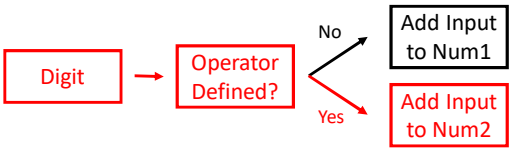


Input parsing strategy

Computer Work

Example: 23+15=

Num1: 23
Num2: 1
Operator: +

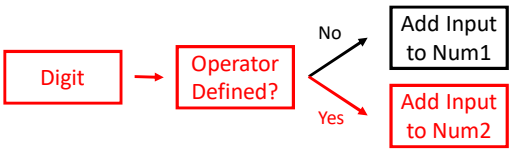


Input parsing strategy

Computer Work

Example: 23+15=

Num1: 23
Num2: 15
Operator: +

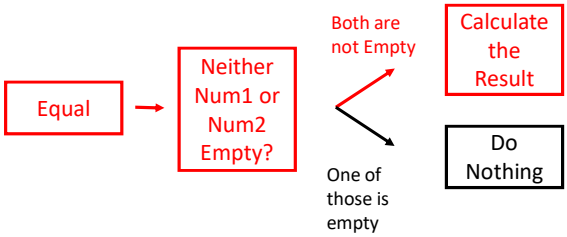


Input parsing strategy

Computer Work

Example: 23+15=

Num1: 23
Num2: 15
Operator: +



How to Calculate in Arduino?

- Use toInt() to convert string to integer.
  Ex. A = num1.toInt(); B = num2.toInt();
- Use switch-case statement to identify the operator
  Ex.

```
switch(operator)
{
  case '+' : C = A + B; break;
  case '-' :  …
    …
}
```

Exercise 3

- ILI9341_XPT2046_Calculator_ESP8266.ino
- Follow the input parsing strategy to complete the code

Input is a digit

```
130  if (lastchar >= '0' && lastchar <= '9' && !toLong)
131  {
132    //If input is digit & operation is not defined
133    if (operation == ' ')
134    {
135      key1 += lastchar;
136      tft.print(lastchar);
137      Serial.println(key1);
138    }
139
140    //If input is digit & operation is defined
141    else
142    {
143      /*TODO*/
144        //You need to determine how key1, key2, operation will change in every TODO.
145        //Write your code here.
146      /*TODO*/
147      tft.print(lastchar);
148      Serial.println(key2);
149    }
```

23

23+1

Input is an operator

```
165      //If input is an operator and operation is not defined
166      else
167      {
168        /*TODO*/
169          //You need to determine how key1, key2, operation will change in every TODO.
170          //Write your code here.
171        /*TODO*/
172        tft.print(operation);
173      }
```

23+

```
{
    operation = lastchar;
    tft.print(operation);
}
{
    key2 += lastchar;
    tft.print(lastchar);
    Serial.println(key2);
}
```

Exercise 3

- ILI9341_XPT2046_Calculator_ESP8266.ino
- Use switch-case statement to complete the calculation function.

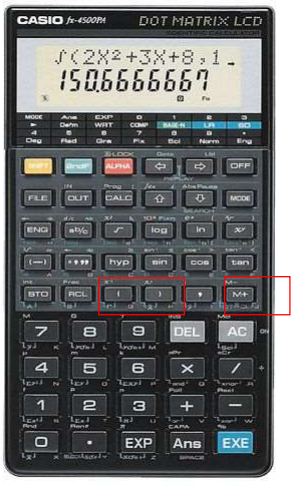
```
400 long int calc(long int num1, long int num2, char op)
401 {
402   /*TODO*/
403     //Write your code here.
404     //This part is to calculate the answer
405     //You can refer to the example in the slide.
406   /*TODO*/
407 }
```

```
long int calc(long int num1, long int num2, char op)
{
 switch (op) {
  case '+':
   return num1 + num2;
  case '-':
   return num1 - num2;
  case 'X':
   return num1 * num2;
  case '/':
   return num1 / num2;
 }
}
```

Bonus Exercise 3
Complex Calculator

- To calculate 2 – 1 * 3 + 5=
- Can be calculated by

2-(1*3)+5=



More Complex Calculator

- Use “postfix”
- Use “stack”

Postfix vs. Infix

- Infix: operator between its operands.
  Ex. 2 + 3, 5 * 4, 2 – 1 * 3 + 5
- Postfix: operator precede its operands.
  Ex. 2 3 +, 5 4 *, 2 1 3 * - 5 +

Postfix vs. Infix

- Infix:
  - Easy for human reading.
  - Hard for program parsing.
- Postfix:
  - Hard for human reading.
  - Easy for program parsing.

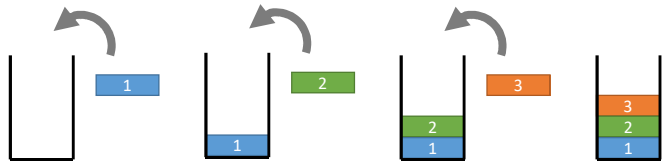
Infix, Prefix, Postfix

Infix Expression	Prefix Expression	Postfix Expression
(A + B) * C	* + A B C	A B + C *

Infix Expression	Prefix Expression	Postfix Expression
A + B * C + D		
(A + B) * (C + D)		
A * B + C * D		
A + B + C + D		

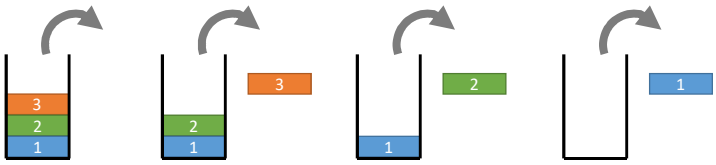
Stack

- A Last-in-first-out data structure

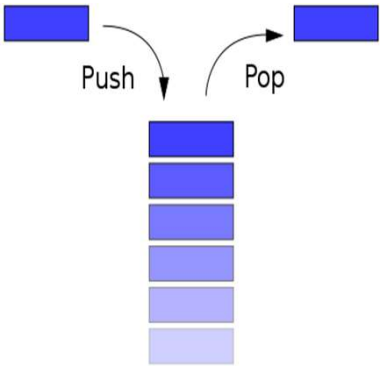


Stack

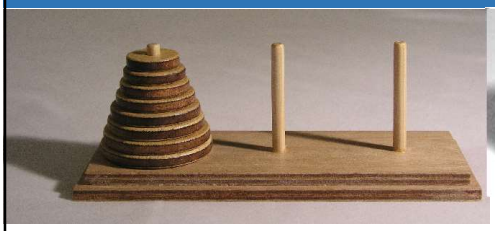
- A Last-in-first-out data structure



Stack



Stack – Application: Tower of Hanoi

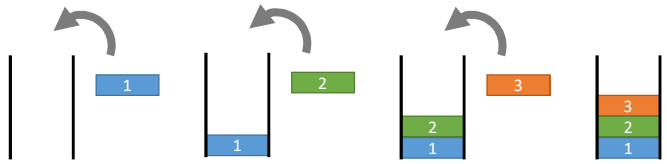


```
#include <stdio.h>
#include <stack>
using namespace std;

int main() {
  int n;
  int i, j, k;
  while(scanf("%d", &n) == 1) {
    stack<int> stk[3];
    for(i = n; i >= 1; i--)
      stk[0].push(i);
    char peg[] = {'A','B','C'};
    if(n&1)   swap(peg[1], peg[2]);
    while(true) {
      int disk[3];
      for(i = 0; i < 3; i++)
        disk[i] = stk[i].size() ? stk[i].top() : 0xffffffff;
      int mnIdx = 0;
      for(i = 1; i < 3; i++)
        if(disk[i] < disk[mnIdx])
          mnIdx = i;
      printf("Move disk %d from %c to %c\n", disk[mnIdx], peg[mnIdx], peg[(mnIdx+1)%3]);
      stk[mnIdx].pop();
      stk[(mnIdx+1)%3].push(disk[mnIdx]);
      if(stk[1].size() == n || stk[2].size() == n)
        break;
```

```
      for(i = 0; i < 3; i++)
        disk[i] = stk[i].size() ? stk[i].top() : 0xffffffff;
      if(disk[0] > min(disk[1], disk[2]) && disk[0] < max(disk[1], disk[2])) {
        if(disk[1] < disk[2]) {
          printf("Move disk %d from %c to %c\n", disk[0], peg[0], peg[2]);
          stk[0].pop(), stk[2].push(disk[0]);
        } else {
          printf("Move disk %d from %c to %c\n", disk[0], peg[0], peg[1]);
          stk[0].pop(), stk[1].push(disk[0]);
        }
      } else if(disk[1] > min(disk[0], disk[2]) && disk[1] < max(disk[0], disk[2])) {
        if(disk[0] < disk[2]) {
          printf("Move disk %d from %c to %c\n", disk[1], peg[1], peg[2]);
          stk[1].pop(), stk[2].push(disk[1]);
        } else {
          printf("Move disk %d from %c to %c\n", disk[1], peg[1], peg[0]);
          stk[1].pop(), stk[0].push(disk[1]);
        }
      } else {
        if(disk[0] < disk[1]) {
          printf("Move disk %d from %c to %c\n", disk[2], peg[2], peg[1]);
          stk[2].pop(), stk[1].push(disk[2]);
        } else {
          printf("Move disk %d from %c to %c\n", disk[2], peg[2], peg[0]);
          stk[2].pop(), stk[0].push(disk[2]);
        }
      }
    }
  }
  return 0;
}
```

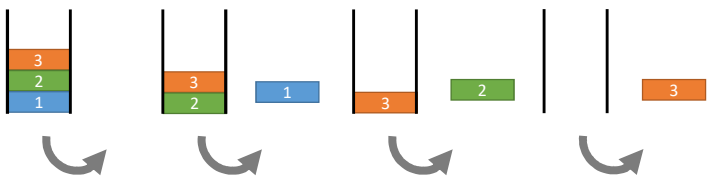
Queue

- A first-in-first-out data structure

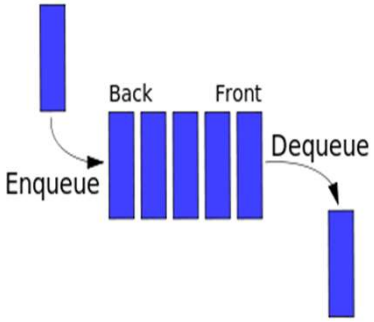


Queue

- A first-in-first-out data structure



Queue



How a Complex Calculator Work? (Bonus Exercise 3)

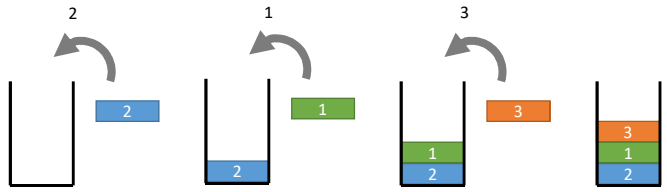
- 2 – 1 * 3 + 5
- Infix → Postfix (Use stack)
- 2 1 3 * - 5 +

Algorithm for Conversion Of An Expression From Infix to Postfix

```
Consider -
    Stack S
    Char  ch
    Char  element
while(Tokens are Available)
 {
     ch = Read(Token);
     if(ch is Operand)
       {
       Print ch ;
       }
     else
       {
       while(Priority(ch) <= Priority(Top Most Stack))
            {
            Print(ele);
            }
       }
}
while(!Empty(S))
{
Print(ele);
}
```

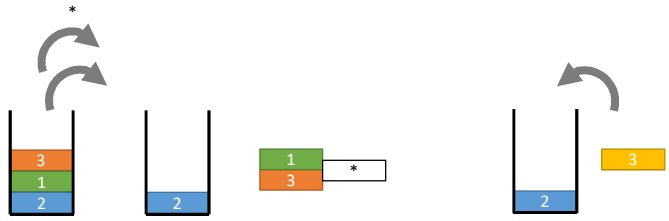
How a Complex Calculator Work?

- (2 1 3 * -) 5 +



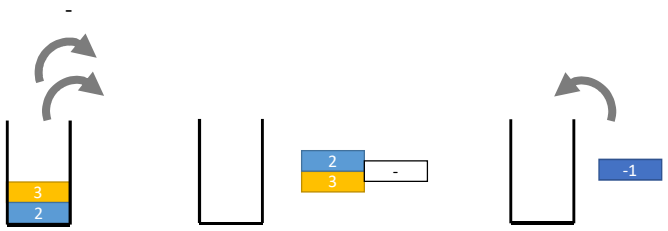
How a Complex Calculator Work?

- 2 1 3 * - 5 +



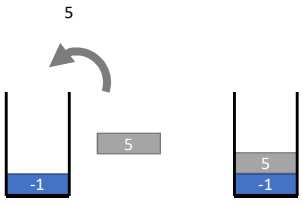
How a Complex Calculator Work?

- 2 1 3 * - 5 +



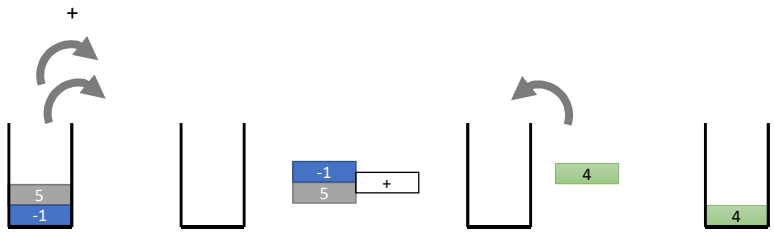
How a Complex Calculator Work?

- 2 1 3 * - 5 +



How a Complex Calculator Work?

- 2 1 3 * - 5 +



From one digit to multiple digits calculation

Using char array
(one digit calculation)

'2'	'-'	'1'	'X'	'3'	'+'	'5'

contain char and int

25	'-'	13	'X'	12	'+'	65

Using int array
(multiple digits calculation)

25	-300	13	-400	12	-200	65

-200 means '+' -300 means '-' -400 means 'X' -500 means '/'

Convert multiple digits to one integer number

'6' '2' '5'

(1) num=0

input digit '6' then num = num*10 + digit (num= 0 * 10 + 6)

(2) num=6

input digit '2' then num = num*10 + digit (num= 6 * 10 + 2)

(3) num=62

input digit '5' then num = num*10 + digit (num= 62 * 10 +

num=625

Bulls and Cows

Introduction

Introduction

Introduction

- Originated from the board game “Mastermind”, which a player arranges colored balls, and the other player guesses the arrangement with information of number of “both correct color and position” and “correct color but wrong position”.
- In Bulls and Cows, colors are replaced by digits 0~9, and no repetition is allowed.
- A : Count of “both correct digit and position”.
- B : Count of “correct digit but wrong position”.

Introduction

Example

Correct answer is “9305”.

- Guess “9034” will result in “1A2B”
  - “9”: correct digit and position
  - “0”&“3”: in the target but in different position.
- Similarly, “1278” result in “0A0B”
- “3905” => “2A2B”
- “1135” => invalid, as ”1” repeats

	Guess	Result
1	1234	0A1B
2	5678	0A1B
3	9012	1A1B
4	9087	1A1B
5	1087	0A1B
6	9205	3A0B
7	9305	4A0B

Program structure


The main structure of the program is divided into 3 parts:

1. Generate a valid random target.

2. Let the player inputs a guess (make sure the validity of the guess).

3. Calculate the result of the guess (i.e. numbers of A and B)

- If the guess is not correct(A is not 4), repeat step 2. Else, the game is over.

Data structure


9462	Digit	0	1	2	3	4	5	6	7	8	9
	Exist	0	0	1	0	1	0	1	0	0	1

- Game data are stored under “gameData” object.
- R and guess are C strings for target and guess.
- histR and histG are 2 boolean array to record whether a digit is in the target and guess respectively.
- count is the number of guesses
- A and B are literally A and B

```
struct GameData {
    char R[5];
    char guess[5];
    bool histR[10];
    bool histG[10];
    int A;
    int B;
    int count;
} gameData;
```

Generate the target

Function “void generate()”

- The function will be called on the initialization of a game
- Use time as the random seed “randomSeed(millis());”
- “random(0, 10)” randomly generate integer 0~9
- Check whether the number is used by “gameData.histR”
- Convert the nubmer to ASCII and append it to the target C string (“gameData.R”)

More on random

- “randomSeed(seed)”: This function initializes the pseudo-random number generator, causing it to start at an arbitrary point in its random sequence, which is very long, random, and is always the same.
- “millis()”: Returns the number of milliseconds since the Arduino board began running the current program.
- “random(max)” or “random(min, max)”: Returns a random number.
  Parameters: min is the lower bound of the random value, inclusive. max min is the upper bound of the random value, exclusive.

ASCII

- American Standard Code for Information Interchange
- Represent control character and printable character with binary 000 0000 to 111 1111
- Control characters: NULL, new line, tab …
- Printable characters: alphabets, numbers, symbols
- ‘0’ is 48 (011 0000) in ASCII
- ‘0’ to ‘9’ is consecutive in ASCII
- ‘0’ + x to convert digit x to ASCII character

ASCII

USASCII code chart

Bits b_7 b_6 b_5 →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 b_3 b_2 b_1 ↓ Row / Column →	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

Generate the target (TODO 01)


Source code:

```
for(int i = 0; i < 4; ++i) {
   gameData.R[i] = ' ';   // use white space to clean
}
int i = 0;
int r;
randomSeed(millis());   // set random seed by time
while (i < 4) {   // four numbers
   r = random(0, 10);   // 0~9
   if (!gameData.histR[r]) {   // non-repeat
      gameData.R[i] = r + '0';  // r + '0', in order to convert to ascii code
      gameData.histR[r] = 1;   // record input number
      ++i;
   }
```

Player Input


Function “void playerMove()”

- When the screen is touched, identify corresponding number or action to the touch.
- Have to prevent invalid (i.e. repeated) input.
- Refresh to display current input guess on the screen.
- Add a small delay in the loop to prevent WDT reset and continuous input.

Player Input

- The UI part is already implemented
- The remaining part is to check for repeated input and append the input to “gameData.guess” 1234
- Use “gameData.histG” to check for repetition 0,1,1,1,1,0,0,0,0,0
- Append ”count” if the input is valid, so the function returns when “count” is 4 (a full guess is made)
- Remember to refresh the display (call “drawGameScreen()”)

Player Input (TODO 02)

Source code:

```
// please refer to playerMove() for details
gameData.guess[count] = '0' + i;
gameData.histG[i] = 1;
++count;
drawGameScreen();
```

Check the Answer

Function “void checkGuess()”

- Calculate A, B, and store them in “gameData.A” and “gameData.B” respectively
- Calculating A is simple
- Can calculate B with the help of “gameData.histG” and “gameData.histR”
- Game over if A equals to 4 (already implemented)

Check the Answer (TODO 03)

Source code:

```
gameData.A = 0;
gameData.B = 0;
for(int i = 0; i < 4; ++i) {
   if(gameData.R[i] == gameData.guess[i]) ++gameData.A;
}
for(int i = 0; i < 10; ++i) {
   if(gameData.histR[i] && gameData.histG[i]) ++gameData.B;
}
gameData.B -= gameData.A;
```

A:
1357
1234

B:
0,1,0,1,0,1,0,1,0,0
0,1,1,1,0,0,0,0,0

Bonus


If there is still some time, try to read and understand the code and implement some additional feature.

- Backspace input.
- Backdoor for the answer.
- Whatever you can come up with.

Bulls and Cows

AI guesses

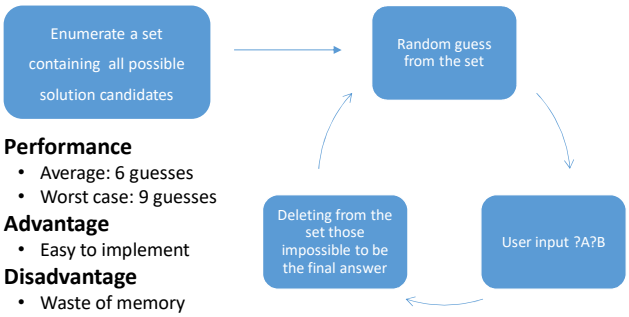

Introduction

Introduction

- Following from last course, we will try to make GameBaby figure out the correct number this time, instead of providing “A” or “B”.
- GameBaby will first display a guess number, and player will need to provide the count of “A” & “B” associated to the correct answer.
- GameBaby will keep on displaying new guess number until player input “4A”.
- If it is impossible to find correct answer, GameBaby will display "Impossible!" and then finish guessing.

Program structure

- **Brutal force**



- **Performance**
  - Average: 6 guesses
  - Worst case: 9 guesses
- **Advantage**
  - Easy to implement
- **Disadvantage**
  - Waste of memory

AIMove

Function “void AIMove()”

- In this function, AI will provide the answer it guesses.
- Generate a random number between 0~(AIData.remain-1), name it “index”.
- Choose a number as “guess number” from “remaining numbers”, according to “index”.

AIMove (TODO 01)

Source code:

```
int index = random(0, AIData.remain);
int counter = 0;
for(int i = 0; i < 10000; ++i) {
    if(AIData.possible[i]) ++counter;
    if(counter > index) {
        AIData.guess = i;
        break;
    }
}
```

- Index is an integer between 0 ~ AIData.remain-1.
- AIData.guess is the guess that AI made, it will be (Index+1)th possible number.

AIFilter

Function “void AIFilter()”

- This function use “latest guess number” and “player input” as reference to remove impossible answers.
- For example, AI first guesses “1234”, player inputs “2A2B”. Therefore, we can remove numbers like “5678”, “0789”…, keep the numbers that have “2A2B” relations with “1234”, like “2134”, “1432”…

AIFilter (TODO 02)


- Preprocess player input data

Source code:

```
bool histG[10] = {0};
  int guess[4] = {0};
  int temp = AIData.guess;
  for(int i = 0; i < 4; ++i) {
     guess[i] = temp % 10;
     histG[temp % 10] = 1;
     temp /= 10;
  }
```

Example:

```
temp = 1357
        %10 ---> 7
temp /= 10 ---> 135
        %10 ---> 5
temp /= 10 ---> 13
        %10 ---> 3
temp /= 10 ---> 1
        %10 ---> 1
temp /= 10 ---> 0
```

AIFilter (TODO 02)


- Remove number that has different “A” or “B”.

Source code:

```
for(int i = 0; i < 10000; ++i) {
  if(AIData.possible[i]) {
     bool histR[10] = {0};
     int target[4] = {0};
     int temp = i;
     int A = 0;
     int B = 0;
     for(int j = 0; j < 4; ++j) {
        target[j] = temp % 10;
        histR[temp%10] = 1;
        temp /= 10;
     }
     for(int j = 0; j < 4; ++j) {
        if(guess[j] == target[j]) ++A;
     }
     for(int j = 0; j < 10; ++j) {
        if(histG[j] && histR[j]) ++B;
     }
     B -= A;
     if((A != AIData.A) || (B != AIData.B)) {
        // remove this number from possible list
        AIData.possible[i] = 0;
        --AIData.remain;
     }
  }
}
```

resetAIData

Function “void resetAIData()”

- The function generates a new possible list of answers.
- For simple, we generate 0~9999 at first, and then remove invalid numbers(i.e. 5566, 1213…)
- There are 5040 remaining possible numbers at begining.

$$10! \div 6! = 10 \times 9 \times 8 \times 7 = 5040$$

resetAIData (TODO 03)

Source code:

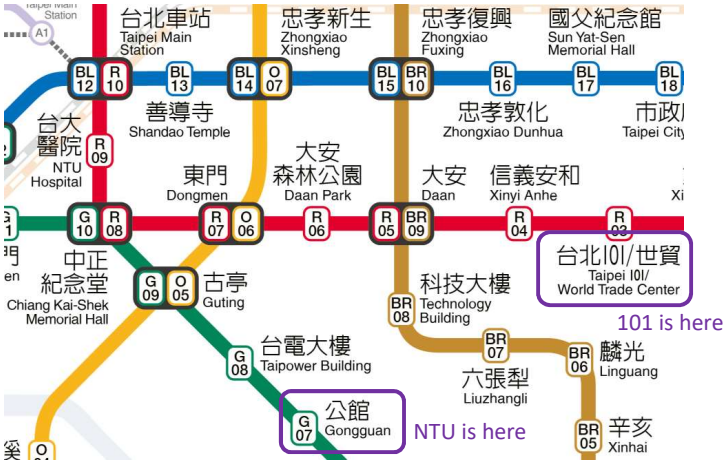
```
for(int i = 0; i < 10000; ++i) {
    AIData.possible[i] = 1;    // add into possible list
    bool hist[10] = {0};
    int target = i;   // example: 1134
    for(int j = 0; j < 4; ++j) {
        int digit = target % 10;
        if(hist[digit]) {  // digit is already exist
            // remove from possible list
            AIData.possible[i] = 0;
            --AIData.remain;
            break;
        }
        hist[digit] = 1;
        target /= 10;
    }
}
```

A* Search

Goal: 8-Puzzle Game

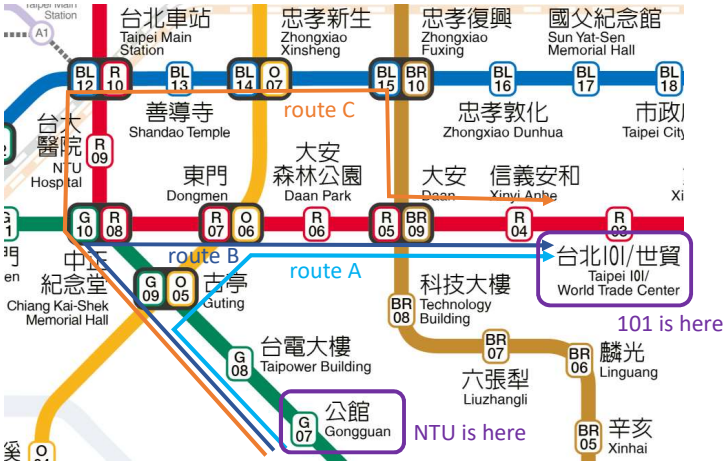
A Trip in Taipei

- Supposed that you want to take the MRT from NTU to Taipei 101
- The fee is NTD 25



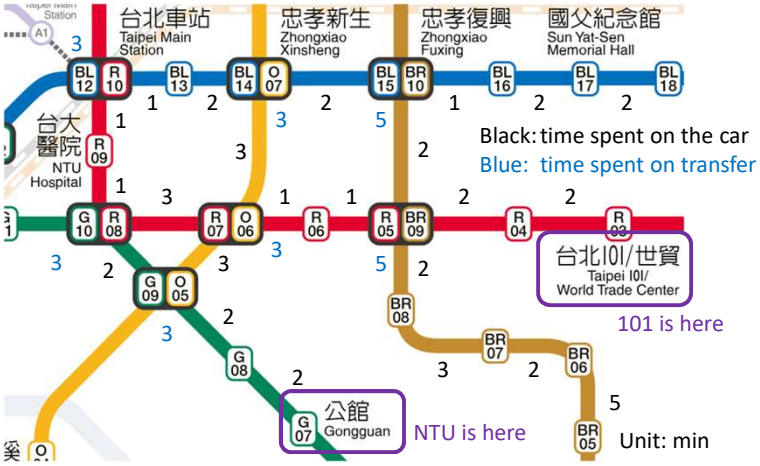
A Trip in Taipei

- There are several ways to go to Taipei 101
- Which route will you choose?



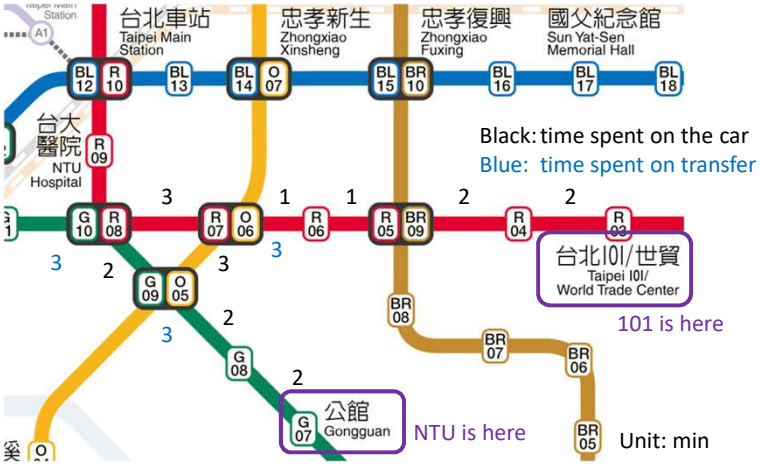
A Trip in Taipei

- After this information is given, which route will you choose?
- (The data are not the real values)



Comparison of Routes

- Route A: 2+2+3+3+3+1+1+2+2 = 19 min
- Route B: 2+2+2+3+3+1+1+2+2 = 18 min (winner)

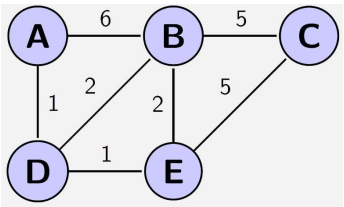


Graph Search Algorithm

- Graph search aims at finding out a path that connects the initial node to the goal node
  - In this case: initial node = NTU, goal node = Taipei 101
- There are several solutions
  - In this case: route A, route B, route C, …
- To analyze whether the path is the optimal solution, we use cost function
  - In this case: time
- The path with the least cost is the optimal solution
  - In this case: route B
- How do we implement graph search by programming?

Terminology of Graph Search

- Initial state: Where you want to start the search
  - For the following example, we use A
- Goal state: Which node you want to reach
  - For the following example, we use C
- Frontier set: Set of the “highlighted nodes” which have not been visited
- Explored set: Set of the nodes which are visited

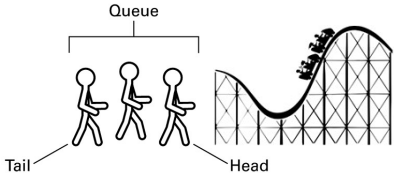


Graph Search Algorithm

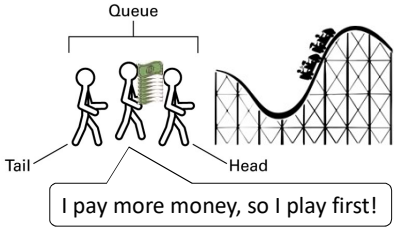

```
1.  initialize the frontier set and push initial state into it
2.  initialize the explored set to be empty
3.  repeat
4.      if the frontier set is empty
5.          return failure
6.      choose a node (say N) in the frontier set
7.      remove N from the frontier set
8.      if N is the goal state
9.          return the corresponding solution
10.     add N to the explored set
11.     list all neighbor nodes of N
12.     if any of the neighbor nodes (say a) are not in the explored set
13.         if a is in frontier set and its newer cost is smaller
14.             update a in frontier
15.         else if a is not in frontier set
16.             add a to frontier
```

Quick Recap: Data Structure

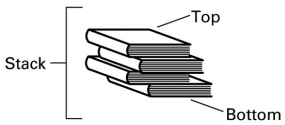
- Queue (first in first out)



- Priority queue (cost dependent)

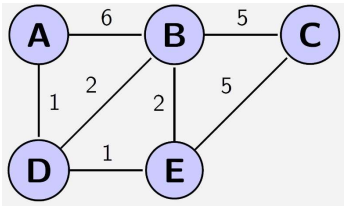


- Stack (last in first out)



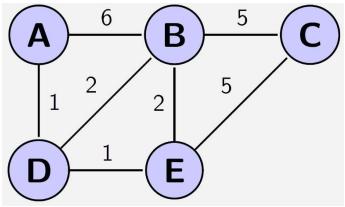
Breadth-First Search (BFS)

- Line 6: Choose a node (say *N*) in the frontier set
  - In BFS, we choose the node that add to frontier set first (queue-like)
- Frontier set:
  - A (initial) → choose A → push B, D
  - B, D → choose B → push C, E
  - D, C, E → choose D → push nothing
  - C, E → choose C → C is goal state
- The solution is A→B→C
- Is it the optimal solution?



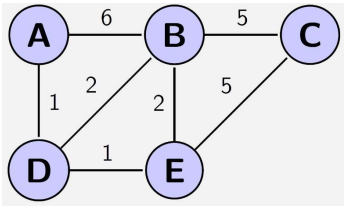
Depth-First Search (DFS)

- Line 6: Choose a node (say *N*) in the frontier set
  - In DFS, we choose the node that add to frontier set first (stack-like)
- Frontier set:
  - A (initial) → choose A → push B, D
  - B, D → choose B → push C, E
  - C, E, D → choose C → C is goal state
- The solution is A→B→C
- Is it the optimal solution?



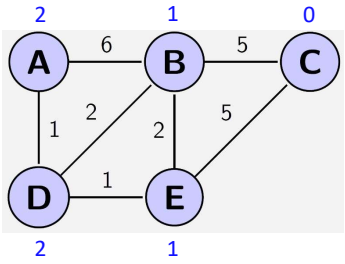
Uniform Cost Search (UCS)

- Line 6: Choose a node (say *N*) in the frontier set
  - In UCS, we choose the node whose summation of cost is minimum (priority-queue-like)
- Frontier set:
  - A (initial) → choose A → push B(0+6), D(0+1), update nothing
  - B(6), D(1) → choose D → push E(1+1), update B(1+2)
  - B(3), E(2) → choose E → push C(2+5), update nothing
  - B(3), C(7) → choose B → push nothing, update nothing
  - C(7) → choose C → C is goal state
- The solution is A→D→E→C
- Is it the optimal solution?



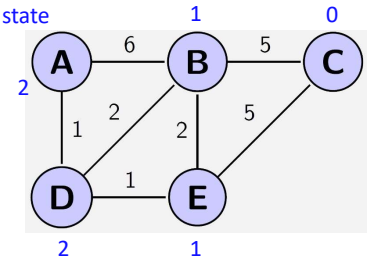
Greedy Search (GS)

- Line 6: Choose a node (say *N*) in the frontier set
  - In GS, we choose the node whose distance to the goal state is the shortest
  - Define distance = the fewest nodes you need to pass to the goal state
- Frontier set:
  - A (initial) → choose A → push B(1), D(2), update nothing
  - B(1), D(2) → choose B → push E(1), C(0), update nothing
  - D(2), E(1), C(0) → choose C → C is goal state
- The solution is A→B→C
- Is it the optimal solution?



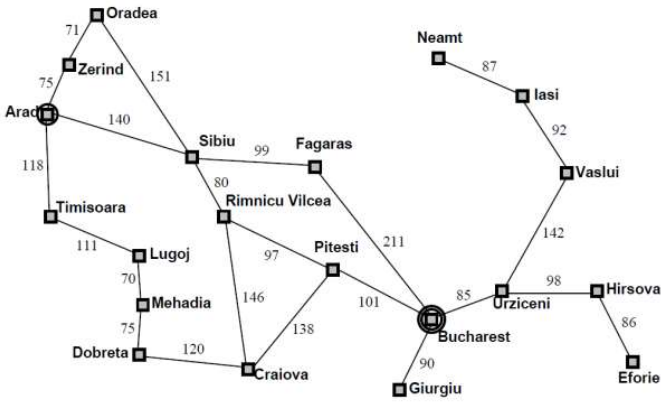
A* Search

- Line 6: Choose a node (say *N*) in the frontier set
  - In A* search, we combine heuristic with real cost to get the final cost
  - Heuristic = the distance defined on the previous page
- Frontier set:
  - A (initial) → choose A → push B(0+6), D(0+1), update nothing
  - B(6+1), D(1+2) → choose D → push E(1+1), update B(1+2)
  - B(3+1), E(2+1) → choose E → push C(2+5), update nothing
  - B(3+1), C(7+0) → choose B → push nothing, update nothing
  - C(7+0) → choose C → C is goal state
- The solution is A→D→E→C
- Is it the optimal solution?

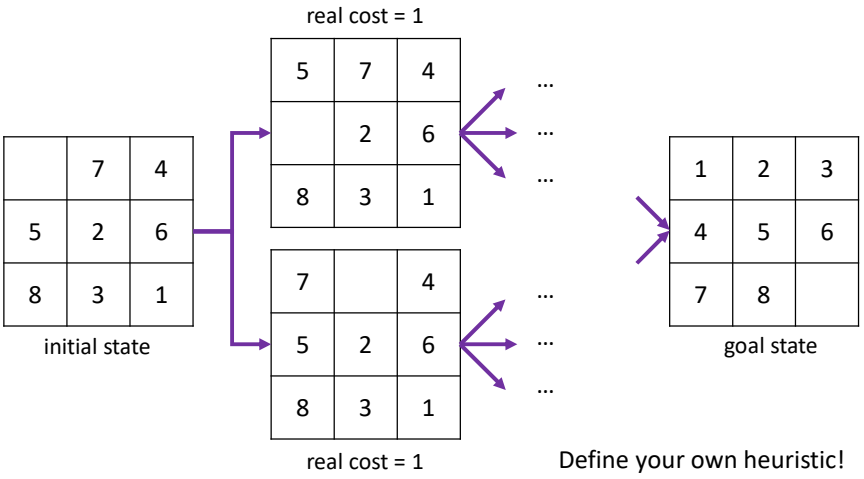


Comparison between UCS and A*

- A* avoids expanding paths that already cost expensively
- Assume that one wants to go to Bucharest from Arad, and the linear distance is the heuristic



Lab: 8-puzzle



Define your own heuristic!
Use A* search

Friendly Hint: Pseudocode


```
1.  initialize the frontier set and push initial state into it
2.  initialize the explored set to be empty
3.  repeat
4.      if the frontier set is empty
5.          return failure
6.      choose a node (say N) in the frontier set
7.      remove N from the frontier set                    Use top(), pop()
8.      if N is the goal state
9.          return the corresponding solution
10.     add N to the explored set                         Use push()
11.     list all neighbor nodes of N
12.     if any of the neighbor nodes (say a) are not in the explored set
13.         if a is in frontier set and its newer cost is smaller
14.             update a in frontier
15.         else if a is not in frontier set
16.             add a to frontier                         Use update(), push()
```

Lab Answer 1

- Code A_star.ino line 434

```
// TODO: Pop state from frontier & update explored and frontier

// Update explored set and frontier set
State s = frontier.top();
explored.push(s);
frontier.pop();
```

Lab Answer 2

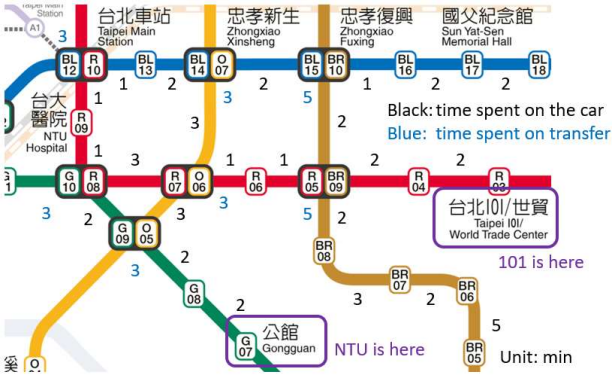
- Code A_star.ino line 466 - 476
- Code A_star.ino line 478- 484

```
if (frontier.find(successor)) {
    /* TODO
     * If successor is already in frontier,
     * (BFS/GS/A*) check if successor's g/h/f is smaller.
     * If so, remove successor in queue & re add it, update frontier and prevStep
     * You can directly compare 2 states' cost by using "<" operator
     * i.e. state1 < state2 returns true if state1's f is smaller than state2's f
     */
    if (successor < frontier.getState(successor)) {
        prevStep.update(successor, step);
        frontier.update(successor);
    }
} else {
    /* TODO
     * Add successor in frontier, update frontier and prevStep
     */
    prevStep.update(successor, step);
    frontier.push(successor);
}
```

Heuristic Design

- Heuristic is a function whose input is the interested state and whose output is a value representing the "estimated cost to goal state"
- For the previous MRT example (the path that minimizes the number of stations one passes):

Station	h
G07	7
G08	6
G09/R08	5
R07/O06	4
R06	3
R05/BR09	2
R04	1
R03	0
…	…



Heuristic Design

- For 8-puzzle:
  - Input state S and get its heuristic value
  - For example, heuristic(S) = difference between S and goal state
  - heuristic(S1) = 8
  - heuristic(S2) = 5
  - Is there a better heuristic to estimate the "cost to goal state" ?
- Special case:
  - If heuristic(state) = 0 for all states, then A* search is the same as UCS

	7	4
5	2	6
8	3	1

State S1

1	2	3
4	5	6
7	8	

Goal State

1	2	3
5	6	8
4	7	

State S2

1	2	3
4	5	6
7	8	

Goal State

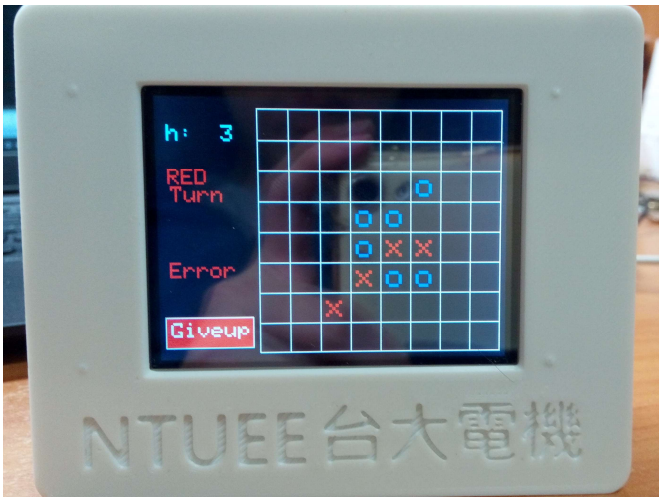
Lab: Comparison on Heuristics


- Change the RANDOM_MOVE from 1 to 13
  - Try default heuristic, how much RANDOM_MOVE can it reach?
  - Try h = 0 (*i.e.* UCS), how much RANDOM_MOVE can it reach?
  - Design your own heuristic, how much RANDOM_MOVE can it reach?
  - Can your heuristic help find the solution before memory error ?

```
// This is calibration data for the raw touc
#define TS_MINX 386
#define TS_MINY 178
#define TS_MAXX 3949
#define TS_MAXY 3906

// TODO: Random step move for initial state
#define RANDOM_MOVE 10
```

Adversarial Search (Minimax)


Goal: Reversi



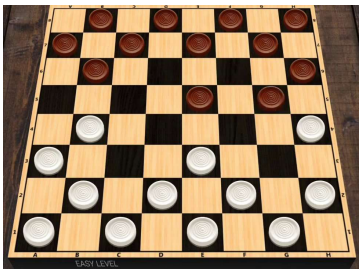
Type of Games

- Adversarial search considers multi-agent and competitive environments.
- Game theory consider both competitive and cooperative environments.
- Most common games are deterministic, turn-taking, two-player, zero-sum games with perfect information.
  - Let's focus on this type of games for a while until told otherwise.

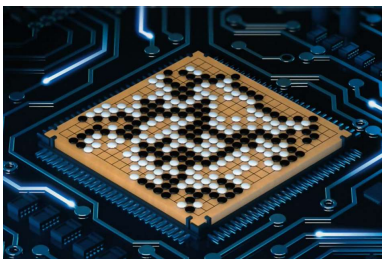
	Deterministic	**Stochastic**
Perfect Information	Chess, Checkers, Go, Othello	Backgammon, Monopoly
Imperfect Information	Battleships, Bingo	Bridge, Poker

Games

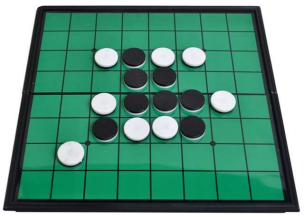

Checkers



Go



Reversi (Othello)



Tic-Tac-Toe

		X
	O	
		O

Playing Tic-Tac-Toe


- Assume that you are playing tic-tac-toe, what is your strategy to win?
- You are X

	X	X
	O	

Playing Tic-Tac-Toe

- If you are a more careful person

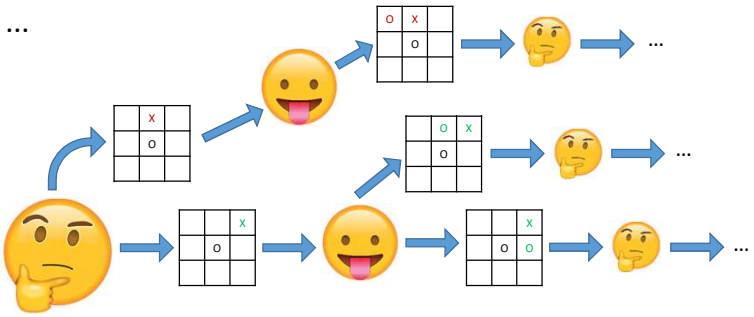
O	X	
	O	

O	X	
	O	
		X

O	X	
O	O	
		X

Playing Strategy

- If it is your turn, you consider on the current step that you have the maximum chance to win
- To define the previous situation, the enemy considers on the next step that you have the minimum chance to win
- To define the previous situation, you consider on the next-next step that you have the maximum chance to win
- …

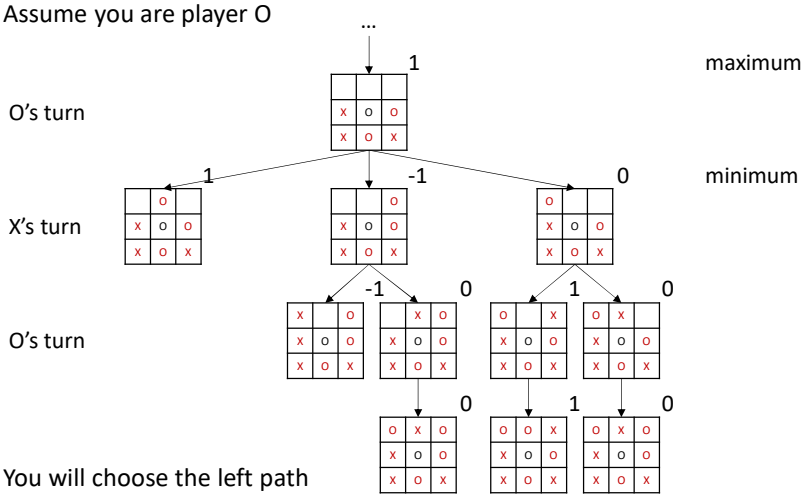


Playing Strategy

- To quantize the value of the end of the game, you use a “utility function”
  - Tic-tac-toe: win(1), tie(0), lose(−1)
  - Reversi: The number of pieces you get minuses that of pieces the enemy gets
- If it is your turn, you find the maximum utility you will get
- If it is the enemy’s turn, you find the minimum utility you will get
- The utility can be found only when you reach the case that the game finishes

Analysis of Utility

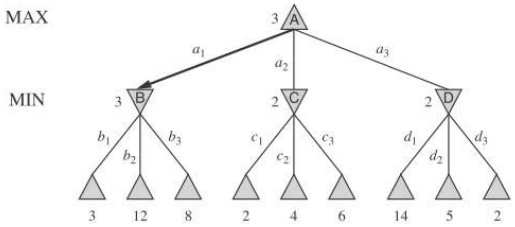
- Assume you are player O



- You will choose the left path

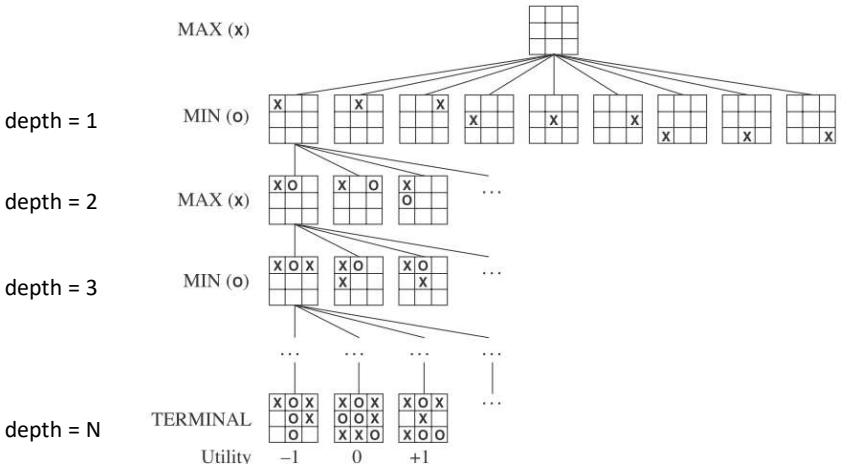
Minimax Search

1. minimax(*s*) // *s* is the interested state
2. **if** *s* is the terminal
3. **return** the corresponding utility
4. **if** *s* is your turn
5. **return** the maximum of minimax(*all possible next steps*)
6. **if** *s* is the enemy's turn
7. **return** the minimum of minimax(*all possible next steps*)
8. end of minimax



Minimax Search

The Real Tic-Tac-Toe Tree

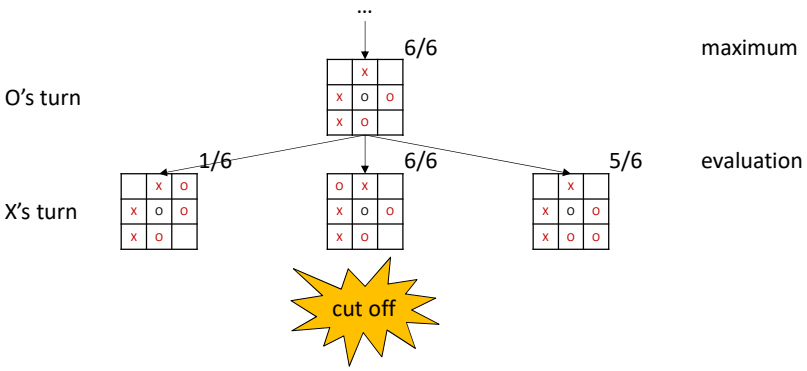


Problems of Minimax Search

- What if this game is reversi? go?
  - Hard to reach terminate state at the preliminary stage of the game
  - You will run out of memory
  - The number of nodes for the state of reversi is about 10^{54}
- Require cutoff and evaluation for non-terminating states
  - Postulate that these states were terminating states

Analysis of Evaluation

- Assume you are player O
- Evaluation: Throw a die to determine the value (1/6~6/6)



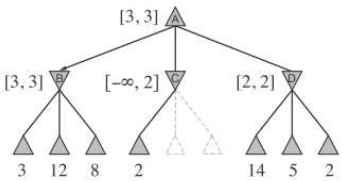
- You will choose the middle path

Modified Minimax Search


```
1.  minimax(s) // s is the interested state
2.      if s is the terminal
3.          return the corresponding utility
4.      if cutoff
5.          return the corresponding evaluation
6.      if s is your turn
7.          return the maximum of minimax(all possible next steps)
8.      if s is the enemy's turn
9.          return the minimum of minimax(all possible next steps)
10. end of minimax
```

Alpha-Beta Pruning

- Not every node needs to be evaluated.

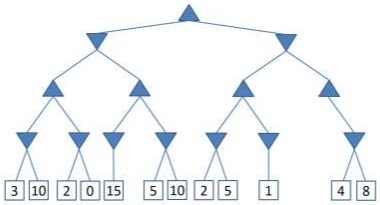


The value of *root* is independent of two unknown nodes x and y

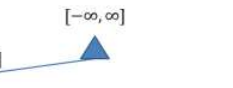
$$\begin{aligned} \textsc{Minimax}(root) &= \max(\min(3,12,8), \min(2,x,y), \min(14,5,2)) \\ &= \max(3, \min(2,x,y), 2)) \\ &= 3 \end{aligned}$$

Alpha-Beta Pruning

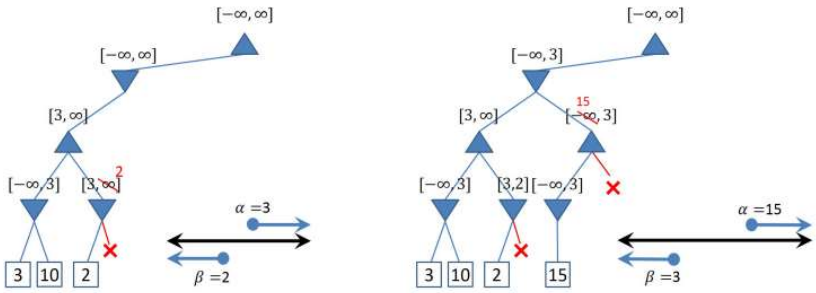
- Keeping α (maximum lower bound) for the maximum utility for player MAX, initialized to $-\infty$.
- Keeping β (minimum upper bound) for the minimum utility for player MIN, initialized to ∞.
- Only the moves within the $[\alpha, \beta]$ window are expanded; otherwise its branches are pruned.
- The pruning does NOT compromise solution quality.



Example



Example

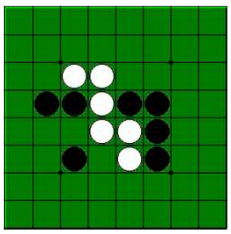


Alpha-Beta Pruning


```
1.  AlphaBeta(s, α, β) // s is the interested state, α/β is lower/upper bound
2.      if s is the terminal
3.          return the corresponding utility
4.      if cutoff
5.          return the corresponding evaluation
6.      if s is your turn
7.          v = −∞
8.          for each step (say a) in the all possible next steps
9.              v = max(v, AlphaBeta(a, α, β))
10.             if v ≥ β return v
11.             α = max(α, v)
12.         return v
13.     else
14.         v = ∞
15.         for each step (say a) in the all possible next steps
16.             v = min(v, AlphaBeta(a, α, β))
17.             if v ≤ α return v
18.             β = min(β, v)
19.         return v
20. end of AlphaBeta
```

Reversi

- **Basic rules**
  Each **reversi piece** has a black side and a white side. On your turn, you place one piece on the board with your color facing up. You must place the piece so that an opponent's piece, or a row of opponent's pieces, is flanked by your pieces. All of the opponent's pieces between your pieces are then turned over to become your color.
- **Aim of the game**
  The object of the game is to own more pieces than your opponent when the game is over. The game is over when neither player has a move. Usually, this means the board is full.



Lab: Design Evaluation Function

- Implement a reversi solver heuristic
- 2 default heuristics is given in **state.h**
  - You may modify weight1, weight2 to create your own heuristic
  - Or you can totally redesign the function **heuristic(State&)**

```
172 double heuristic(State& s) {
173     // Default heuristic
174     bool available[8][8];
175     int redMoves = availablePlaces(s, availab
176     int blueMoves = availablePlaces(s, availa
177
178     double h = 0;
179
180     double h1 = redMoves - blueMoves;
181     double h2 = countResult(s);
182
183     // TODO: design your heuristic fun
184     /**** Write your code here ****/
185     // double weight1 = 0.5;
186     // double weight2 = 0.5;
187     // h = weight1*h1 + weight2*h2;
188
189     /*******************************/
190
191     return h;
192 }
```

```
183     // TODO: design your heuristic functi
184     /**** Write your code here ****/
185     // double weight1 = 0.5;
186     // double weight2 = 0.5;
187     // h = weight1*h1 + weight2*h2;
```

Lab: Design Evaluation Function

- The 2 given naïve evaluation functions are:
  - # of red Xs – # of blue Os
  - # of available moves of red Xs – # of available moves of blue Os
- Structure of State
  - You can use State.pos[i][i] and State.exist[i][j] to access state status
  - You may see the code given to get familiar with the usage

```
struct State {
    bool pos[8][8]; // false: x (red), true: o (blue)
    bool exist[8][8];
};
```

優秀作品賞析—
研究題材的發展

2015 — EBED018I Magnetic Positioning Sphere: A Single-Source 3D Positioning System using Rotating Magnetic Fields

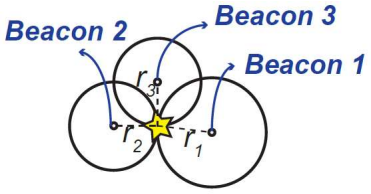
Wei-Tung Chen
Taipei Municipal Jianguo High School, Taipei City, Taiwan This study presents a 3D magnetic positioning system using a single-source Magnetic Positioning Sphere (MPS). The proposed MPS system is composed of three mutually orthogonal coils and an Arbitrary Waveform Generator (AWG) to generate the designated signals. By applying Frequency Division Multiplexing (FDM), the three coils generate two rotating magnetic fields with different frequencies of phase-quadrature current signals, which is equivalent to two mechanically rotating magnetic dipoles. To obtain the accurate position in a 3-dimensional space, the object is equipped with a magnetometer for measuring the strength of magnetic field and a micro-controller unit for signal processing. Specifically, the magnetometer is triggered by a designated pulse, and its measurement result is extracted into two signals of different frequencies. The phases of the two signals are then used to determine the elevation and azimuth angles, and the amplitudes of these signals are used to calculate the distance between the source and the object. Using a comprehensive set of experiments, we demonstrate that the proposed MPS system can achieve an accuracy within 5 cm. Compared to the existing RSSI-based indoor positioning systems, the MPS system has no need of doing site surveys, which makes it easy for quick deployment. Moreover, the MPS system is based on magnetic field, which is robust against radio interference and environment obstructions. The proposed MPS system has great potential to facilitate a wide range of applications, such as mobile systems, wearable computing, and location-based applications, where an instant and accurate indoor 3D positioning system is demanded. ***Awards won at the 2015 Intel ISEF***
First Award of $3,000 Honorable Mention from National Aeronautics and Space Administration

Value Creation Course from SRI

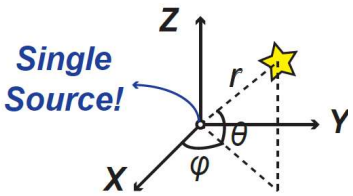
- N(eed)
- A(pproach)
- B(enefit)
- C(ompetition)

摘要
發展框架
檢查清單

磁場定位球



C Fig. 1. Multi-beacon positioning system



N Fig. 2. Single-source positioning system

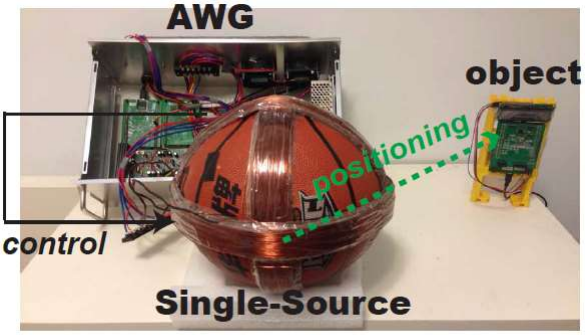


Fig. 11. MPS system prototype

A

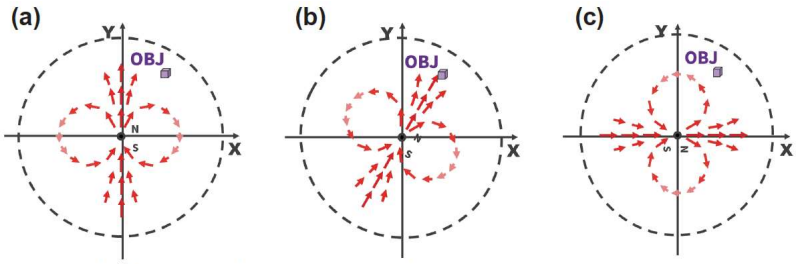


Fig. 4. Magnetic field generated by a rotating magnetic dipole at different timestamps (red arrows are magnetic flux lines)

(a) Field at t_1 (b) Field at t_2 (c) Field at t_3

A

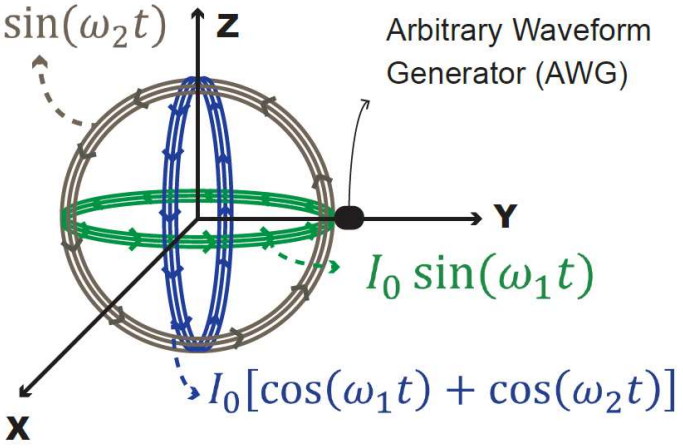
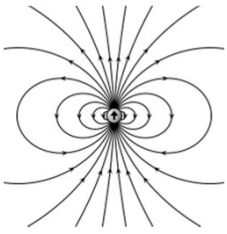
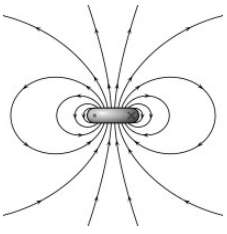


Fig. 8. Magnetic Positioning Sphere (MPS)

Electrical and Magnetic Dipoles





From: Wikipedia
Please check the reference for the solution!

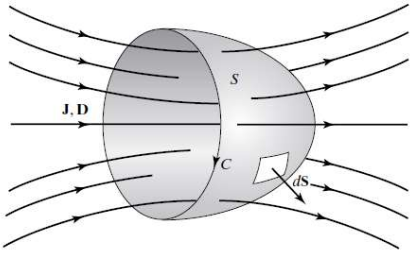
安培定律

$$\oint_C \mathbf{H}\cdot d\mathbf{l} = \int_S \mathbf{J}\cdot d\mathbf{S} + \frac{d}{dt}\int_S \mathbf{D}\cdot d\mathbf{S}$$

(I_C and I_D)

- *the magnetomotive force around a closed path C is equal to the algebraic sum of the current due to flow of (free) charges and the displacement current bounded by C.*

1. The surface integrals on the right side are to be evaluated in accordance with the R.H.S. rule.
2. In evaluating the surface integrals, *any surface S bounded by C* can be employed. However, *the same surface* must be employed for the two surface integrals.

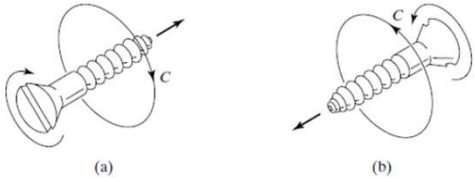


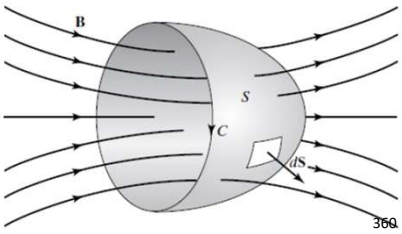
法拉第定律

$$\oint_C \mathbf{E}\cdot d\mathbf{l} = -\frac{d}{dt}\int_S \mathbf{B}\cdot d\mathbf{S}$$

where *S* is a surface bounded by the closed path *C*.
In words, Faraday's law states that *the electromotive force around a closed path is equal to the negative of the time rate of change of the magnetic flux enclosed by that path*.

1. The magnetic flux on the right side is to be evaluated in accordance with the *right-hand screw rule* (R.H.S. rule),





研究報告撰寫的要點

Value Creation Course from SRI

- N(eed)
- A(pproach)
- B(enefit)
- C(ompetition)

摘要

發展框架

檢查清單

Abstract

2015 — EBED018I Magnetic Positioning Sphere: A Single-Source 3D Positioning System using Rotating Magnetic Fields

Wei-Tung Chen

Taipei Municipal Jianguo High School, Taipei City, Taiwan

N A B C This study presents a 3D magnetic positioning system using a single-source Magnetic Positioning Sphere (MPS). The proposed MPS system is composed of three mutually orthogonal coils and an Arbitrary Waveform Generator (AWG) to generate the designated signals. By applying Frequency Division Multiplexing (FDM), the three coils generate two rotating magnetic fields with different frequencies of phase-quadrature current signals, which is equivalent to two mechanically rotating magnetic dipoles. To obtain the accurate position in a 3-dimensional space, the object is equipped with a magnetometer for measuring the strength of magnetic field and a micro-controller unit for signal processing. Specifically, the magnetometer is triggered by a designated pulse, and its measurement result is extracted into two signals of different frequencies. The phases of the two signals are then used to determine the elevation and azimuth angles, and the amplitudes of these signals are used to calculate the distance between the source and the object. Using a comprehensive set of experiments, we demonstrate that the proposed MPS system can achieve an accuracy within 5 cm. Compared to the existing RSSI-based indoor positioning systems, the MPS system has no need of doing site surveys, which makes it easy for quick deployment. Moreover, the MPS system is based on magnetic field, which is robust against radio interference and environment obstructions. The proposed MPS system has great potential to facilitate a wide range of applications, such as mobile systems, wearable computing, and location-based applications, where an instant and accurate indoor 3D positioning system is demanded. ***Awards won at the 2015 Intel ISEF***

First Award of $3,000 Honorable Mention from National Aeronautics and Space Administration

Value Creation Course from SRI

- N(eed)
- A(pproach)
- B(enefit)
- C(ompetition)

摘要

發展框架

檢查清單

Competition, 文獻探討

Background

1. Global Positioning System (GPS) is the ubiquitous system for outdoor positioning, however, it performs very poorly indoors.
2. Received Signal Strength Indicator (RSSI) based techniques commonly used for indoor positioning only carry distance information, and they requires at least three beacon sources to locate an object〔Fig. **1**〕.

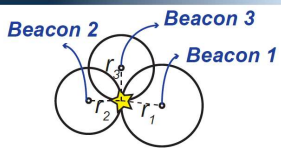


Fig. 1. Multi-beacon positioning system

3. Current methods of magnetic-based positioning, such as pattern matching and DC magnetic field signal strength, require extensive site surveys and sophisticated setup procedures. Yet, an easy-setup system for precise indoor positioning is highly desirable.

Need, 目標

Objectives

Investigation into a new 3D positioning system that can achieve:

1. **single-source positioning** by calculating the distance (r), elevation angle (θ) and azimuth angle (φ) between an object and the reference point〔Fig. **2**〕;
2. precise indoor positioning in **3-dimensional space**; and
3. easy setup without site surveys.

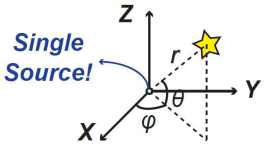


Fig. 2. Single-source positioning system

Approach,原理與方法

Methods

Calculation of Distance (*r*) using Received Signal Strength (RSS)

1. Extreme low frequency time-varying magnetic field signals are sent from the source, and the field strength can be determined by the **amplitude** of the signal.
2. Using quasistatic approximation, a magnetic dipole field model is obtained to tell the relationship between the distance and the magnetic field strength〔Fig. **3**〕, i.e. (μ_0: Magnetic Constant; M : Magnetic Dipole Moment),

$$\boldsymbol{B} = \frac{\mu_0}{4\pi}\frac{M}{r^3}(2\sin\theta\,\hat{\boldsymbol{r}} - \cos\theta\,\hat{\boldsymbol{\theta}}) \qquad (1)$$

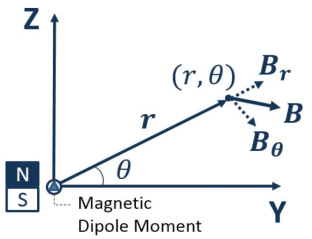


Fig. 3. Magnetic dipole field model

Introduction of the Magnetic Positioning Sphere (MPS)

1. MPS consists of three mutually orthogonal coils and an Arbitrary Waveform Generator (AWG)〔Fig. **8**〕.
2. The AWG sends phase-quadrature signals to the coils to simulate a rotating magnetic field.
3. The Frequency Division Multiplexing (FDM) allows an object to gather two rotating magnetic field signals at different frequencies simultaneously〔Fig. **9**〕.
4. A third signal is sent to the three coils, and it is converted into a magnetic field as a reference to calculate the elevation and the azimuth angles of the object.

Phase Shift Keying (PSK)

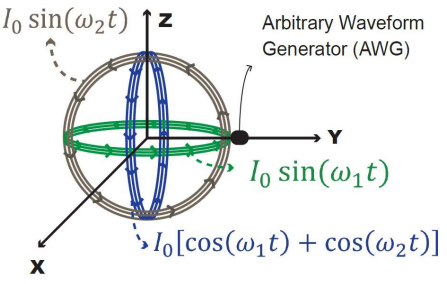


Fig. 8. Magnetic Positioning Sphere (MPS)

Calculations of Elevation Angle (θ) and Azimuth Angle (φ)

1. Concept of a Rotating Magnetic Field

Magnetic field strength received by an object **varies** over time, and the peak field strength occurs when magnetic dipole's north and south poles face the object〔Fig. **4** & **5**〕.

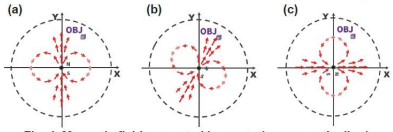


Fig. 4. Magnetic field generated by a rotating magnetic dipole at different timestamps (red arrows are magnetic flux lines)
(a) Field at t_1 (b) Field at t_2 (c) Field at t_3

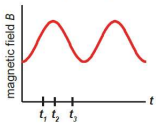


Fig. 5. Magnetic field strength received by an object in a rotation cycle

2. Calculations of Angles using a Rotating Magnetic Field

A. In this study, a rotating magnetic field is created by feeding **phase-quadrature** signals to two mutually orthogonal coils.

B. While focusing 0 to 180 degrees of a rotating cycle, the peak strength of magnetic field occurs at $\tilde{\boldsymbol{t}}$, and the angle of an object is calculated $\tilde{\boldsymbol{\theta}}$, i.e. (ω: the angular frequency of a rotating magnetic field)〔Fig. **6**〕,

$$\tilde{\theta} = \omega\tilde{t} \qquad (2)$$

C. Angle calculations for a complete 360 degrees will be achieved during system development.

D. $\tilde{\boldsymbol{r}}$ is the distance from the origin point to the object.

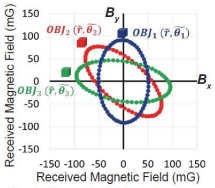


Fig. 6. Magnetic field received over time by three objects at different angles while two axes of each sensor are fixed on X and Y axis

Implementation

1. The MPS is composed of a signal source with three mutually orthogonal coils and an Arbitrary Waveform Generator (AWG), which is composed of amplifiers and an MCU to generate phase-quadrature signals〔Fig. **11**〕.
2. The object being detected uses a magnetometer to receive the rotating magnetic field strength signals.

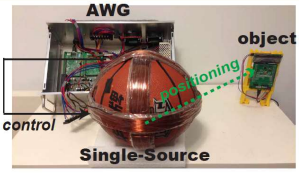


Fig. 11. MPS system prototype

Benefit, 結果討論

Evaluation Results

Positioning Accuracy and Stability of MPS

The experiment results show that the MPS system can achieve accurate indoor positioning at a median error of 5 cm, and the standard deviation of 2〔Fig. **12** & **13**〕.

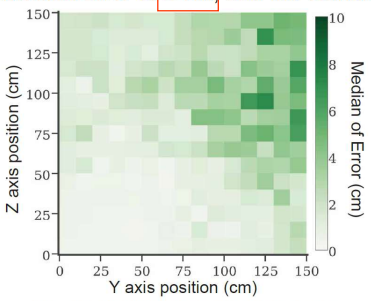


Fig. 12. Median of positioning error

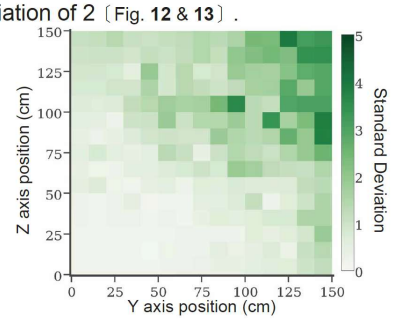


Fig. 13. Standard deviation of positioning error

Positioning Accuracy with Obstructions in Different Zones

1. The positioning accuracy was evaluated with different obstructions in the three zones within a 3D space〔Fig. **14**〕.
2. The results show that wooden obstructions have little to no effect on positioning, while metal obstructions do produce slight errors due to the induction fields; however, they are negligible〔Fig. **15**〕.

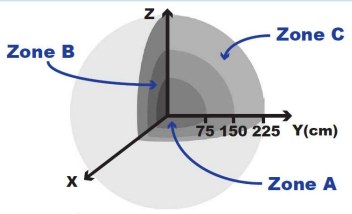


Fig. 14. Definition of zones in a 3D space

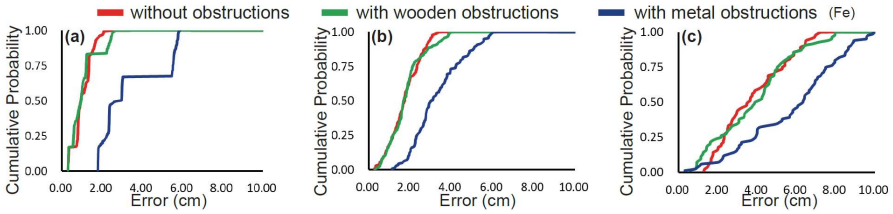


Fig. 15. Cumulative Distribution Function for positioning error with and without obstructions

(a) Error of positioning in **Zone A** (b) Error of positioning in **Zone B** (c) Error of positioning in **Zone C**

Abstract

2015 — EBED018I Magnetic Positioning Sphere: A Single-Source 3D Positioning System using Rotating Magnetic Fields

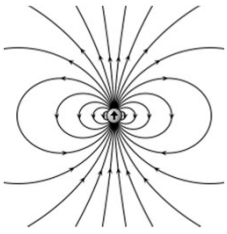
Wei-Tung Chen

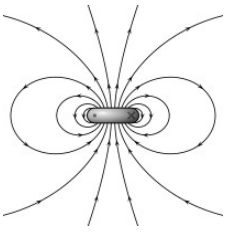
Taipei Municipal Jianguo High School, Taipei City, Taiwan

This study presents a 3D magnetic positioning system using a single-source Magnetic Positioning Sphere (MPS). The proposed MPS system is composed of three mutually orthogonal coils and an Arbitrary Waveform Generator (AWG) to generate the designated signals. By applying Frequency Division Multiplexing (FDM), the three coils generate two rotating magnetic fields with different frequencies of phase-quadrature current signals, which is equivalent to two mechanically rotating magnetic dipoles. To obtain the accurate position in a 3-dimensional space, the object is equipped with a magnetometer for measuring the strength of magnetic field and a micro-controller unit for signal processing. Specifically, the magnetometer is triggered by a designated pulse, and its measurement result is extracted into two signals of different frequencies. The phases of the two signals are then used to determine the elevation and azimuth angles, and the amplitudes of these signals are used to calculate the distance between the source and the object. Using a comprehensive set of experiments, we demonstrate that the proposed MPS system can achieve an accuracy within 5 cm. Compared to the existing RSSI-based indoor positioning systems, the MPS system has no need of doing site surveys, which makes it easy for quick deployment. Moreover, the MPS system is based on magnetic field, which is robust against radio interference and environment obstructions. The proposed MPS system has great potential to facilitate a wide range of applications, such as mobile systems, wearable computing, and location-based applications, where an instant and accurate indoor 3D positioning system is demanded. ***Awards won at the 2015 Intel ISEF***

First Award of $3,000 Honorable Mention from National Aeronautics and Space Administration

Electrical and Magnetic Dipoles





From: Wikipedia
Please check the reference for the solution!

Approach,原理方法

Methods

Calculation of Distance (*r*) using Received Signal Strength (RSS)

1. Extreme low frequency time-varying magnetic field signals are sent from the source, and the field strength can be determined by the **amplitude** of the signal.
2. Using quasistatic approximation, a magnetic dipole field model is obtained to tell the relationship between the distance and the magnetic field strength〔Fig. **3**〕, i.e. (μ_0: Magnetic Constant; *M* : Magnetic Dipole Moment),

$$\boldsymbol{B} = \frac{\mu_0}{4\pi}\frac{M}{r^3}(2\sin\theta\,\hat{\boldsymbol{r}} - \cos\theta\,\hat{\boldsymbol{\theta}}) \qquad (1)$$

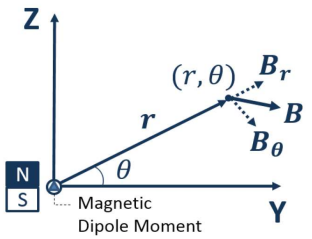


Fig. 3. Magnetic dipole field model

適用性問題?

誤差評估?

物理極限?

Coverage of MPS

1. The magnetic field strength is determined by the value of **NI_0A**, which considers the number of turns of coil; electric current flow (Amperes); and the area of coil (m^2), as illustrated in formula ***(3)*** 〔Table **1(a)**〕.
2. The achieved range of MPS is sensitive to the granularity supported by Analog to Digital Converter (ADC) and the amplification gain of the magnetometer〔Table **1(b)**〕.

Table 1. Coverage of MPS under different conditions

(a) MPS with different NI_0A

NI_0A	Measured range (m)	Ideal range (m)
1.15	1.46	2.64
2.89	1.88	3.59
5.39	2.51	4.57
8.32	2.68	5.12
62.83	5.32	10.15
353.43	N/A	18.06
1963.49	N/A	31.99
3848.45	N/A	40.03

(b) Objects with different sensitivities

NI_0A	Bits of ADC	Gain of sensors	Measured Range (m)
2.89	12 bits	212.77	1.88
		638.30	2.21
	16 bits	212.77	1.95
		638.30	2.29
8.32	12 bits	212.77	2.68
		638.30	3.17
	16 bits	212.77	2.76
		638.30	3.19

Conclusions and Future Applications

1. A novel system, called **Magnetic Positioning Sphere**, is presented as an accurate and single source indoor positioning.
2. The MPS can locate an object is less than 10 ms, and it is resilient against radio interference and obstructions.
3. Compared with current positioning systems, the MPS system does not require a site survey and can be installed instantly with minimal deployment cost.
4. Future applications of the MPS system may include applications which demand an **instant setup** positioning system. For example, it can be installed in seconds and provide positioning of firefighters, robots, or UAVs in rescue situations.

Abstract

2015 — EBED018I Magnetic Positioning Sphere: A Single-Source 3D Positioning System using Rotating Magnetic Fields

Wei-Tung Chen

Taipei Municipal Jianguo High School, Taipei City, Taiwan

This study presents a 3D magnetic positioning system using a single-source Magnetic Positioning Sphere (MPS). The proposed MPS system is composed of three mutually orthogonal coils and an Arbitrary Waveform Generator (AWG) to generate the designated signals. By applying Frequency Division Multiplexing (FDM), the three coils generate two rotating magnetic fields with different frequencies of phase-quadrature current signals, which is equivalent to two mechanically rotating magnetic dipoles. To obtain the accurate position in a 3-dimensional space, the object is equipped with a magnetometer for measuring the strength of magnetic field and a micro-controller unit for signal processing. Specifically, the magnetometer is triggered by a designated pulse, and its measurement result is extracted into two signals of different frequencies. The phases of the two signals are then used to determine the elevation and azimuth angles, and the amplitudes of these signals are used to calculate the distance between the source and the object. Using a comprehensive set of experiments, we demonstrate that the proposed MPS system can achieve an accuracy within 5 cm. Compared to the existing RSSI-based indoor positioning systems, the MPS system has no need of doing site surveys, which makes it easy for quick deployment. Moreover, the MPS system is based on magnetic field, which is robust against radio interference and environment obstructions. The proposed MPS system has great potential to facilitate a wide range of applications, such as mobile systems, wearable computing, and location-based applications, where an instant and accurate indoor 3D positioning system is demanded. ***Awards won at the 2015 Intel ISEF***

First Award of $3,000 Honorable Mention from National Aeronautics and Space Administration