Lab 1 – A Simple IoT Application

Introduction to IoT
Outline

• Objectives
• Basic Raspberry Pi
• OS Installation
• Simple IoT Applications
  • Controlling LED
    • with Raspberry Pi
    • With switch
    • With Light Sensor (LDR)
  • Temperature & Humidity Monitoring
Objectives

• Get to know Raspberry Pi
• Capable to Install the Raspberry Operating System
• Connecting Sensors to Raspberry Pi
• Writing the code to run the sensors
Basic Raspberry Pi – What is a Raspberry Pi?

A low cost, credit-card sized computer

**Processor:** Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz

**Memory:** 2GB, 4GB or 8GB LPDDR4 (depending on model)

**Connectivity:** 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 x USB 3.0 ports 2 x USB 2.0 ports.

**GPIO:** Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
Basic Raspberry Pi – What is a Raspberry Pi?
Basic Raspberry Pi – Connecting Raspberry Pi

• Using Micro HDMI Ports
  • Connect the Micro HDMI into LCD Monitor

• Using Network
  • By Wire / Wireless
  • Connect by using SSH or VNC
Basic Raspberry Pi – Breadboard
Basic Raspberry Pi – Sensors

• What is Sensors
  • add almost-human sensing capabilities
  • taking real-world events
  • converting them to analogue or digital signals
  • Read by Raspberry Pi
Basic Raspberry Pi – Sensors

• Sensor Categories
  • Temperature / Humidity / Air Pressure / Gas
  • Motion Sensors
  • Navigation Modules
  • Wireless / Infrared (IR) / Bluetooth
  • Analogue Sensors
  • Current Supply
  • Displays
  • Other Modules, Components and Sensors
Basic Raspberry Pi – Actuators

• What is Actuators
  • Convert an electrical signal into a corresponding physical quantity
    • Example: movement, force, sound etc.
  • Controlled by Raspberry Pi
Basic Raspberry Pi – Sensors vs Actuators

• Different Sensors and Actuators
  • Sensors : read and get the information from sensors
  • Actuators : write and control some tools based on the previous information
Basic Raspberry Pi – Sensors and Actuators

How to get it?

• Borrow from us
  • Just have the limited sensors and actuators
• Buy it by yourself
  • Save the receipt and reimburse to us
  • The limit of amount to reimburse: 1,000 NTD per team
  • The receipts should show the following title or number
Basic Raspberry Pi – Sensors and Actuators

- Where to buy the sensors and actuators?
OS Installation – things you need at first

Make sure you already get all of them below:

- Raspberry Pi 4 Model B
- USB type-C power supply
- microSD card

Something you also need:

- card reader (for microSD)
- network cable (Ethernet RJ45)
- laptop or PC
OS Installation

• Download Raspbian at here:

• Choose the version you like and unzip the .zip file

• Here I choose **Raspberry Pi OS (32-bit) with desktop and recommended software** since we have 32GB SD card
OS Installation

• Right now pi 4b only support booting from SD card, so we need to download a tool to flash OS image to SD card

• Rufus (Windows only) or balenaEtcher (Windows / macOS / Linux)

• Flash .img file into your SD card
(You need a SD card reader to help you complete this step.)
OS Installation
OS Installation – SSH

• After process completes, add a **new raw file** called “ssh” into the “boot” disk.

Check:
Plug the microSD card into pi 4b, and connect type-C power cable and network cable.

If green light is twinkling under the left corner of network cable slot, that means your pi 4b is using SSH now!!
OS Installation – SSH

• There is a built-in tool for ssh in Windows 10. But if you cannot find it, you need to download PuTTY here.

• Use ssh command
  “ssh pi@raspberrypi.local”
  default password: raspberry

C:\Users\chenj>ssh pi@raspberrypi.local
pi@raspberrypi.local's password:
Linux raspberrypi 4.19.57-v7l+ #1244 SMP Thu Jul 4 18:48:07 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
Last login: Sat Sep 28 21:50:49 2019

SSH is enabled and the default password for the 'pi' user has not been changed. This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~ $
OS Installation – Enable VNC server

• sudo raspi-config

Step1. choose 5 Interfacing Options -> P3 VNC -> Yes (是)
Step2. choose 7 Advanced Options -> A5 Resolution -> choose one other than Default

You need to reboot the system after the setting!
OS Installation – VNC client

• UltraVNC (Windows only) or RealVNC (Windows / macOS / Linux)
• Connect to the VNC Server “raspberrypi.local”
After you configure WiFi connection on Pi 4b, you can use VNC connect to Pi 4b without network cable.

Use command `ifconfig` to find what is the ip address on wlan.
OS Installation - Notes
OS Installation - Notes

If you reinstall the Raspberry Pi again, maybe you will encounter some problem when you using SSH command.

Try to delete the “known_hosts” file or just delete the line related to “raspberrypi.local” in the known_hosts and use ssh command again!
OS Installation - Notes

@ WARNING: POSSIBLE DNS SPOOFING DETECTED!

The ECDSA host key for raspberrypi.local has changed,
and the key for the corresponding IP address fe80:1b0b:8f29:b0b0:b9fc2%7
is unknown. This could either mean that
DNS SPOOFING is happening or the IP address for the host
and its host key have changed at the same time.

@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!

It is possible that someone is doing something nasty!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:Xgkt5/45SrRcKdA2An2y/4gW3Hd1wRYse04YoJTC6.
Please contact your system administrator.
Add correct host key in C:users\chenj\.ssh\known_hosts to get rid of this message.
offending ECDSA key in C:users\chenj\.ssh\known_hosts:11
ECDSA host key for raspberrypi.local has changed and you have requested strict checking.
Host key verification failed.

C:users\chenj\.ssh
.ssh, 您確定要執行嗎 (YN)? Y
Simple IoT Applications
Controlling LED with Raspberry Pi

Components:
• LED
• A Resistor (Orange, Orange, Brown, Gold)

```python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

ledPin = 12
GPIO.setup(ledPin, GPIO.OUT)

for i in range(100):
    print("LED turning on.")
    GPIO.output(ledPin, GPIO.HIGH)
    time.sleep(1)
    print("LED turning off.")
    GPIO.output(ledPin, GPIO.LOW)
    time.sleep(1)
```

There are two different model of GPIO.setmode (pin numbering):
• GPIO.BOARD : using board numbering system (ex: pin 12)
• GPIO.BCM : using BCM numbers (ex: GPIO 18)
Controlling LED with Raspberry Pi
Controlling LED with Switch

Components :
- LED
- Switch
- 1 Resistors (Orange, Orange, Brown, Gold)

```python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(20, GPIO.IN, pull_up_down=GPIO.PUD_UP)  # Button to GPIO20
GPIO.setup(24, GPIO.OUT)  # LED to GPIO24

try:
    while True:
        button_state = GPIO.input(20)
        if button_state == False:
            GPIO.output(24, True)
            print('Button Pressed...')
            time.sleep(0.2)
        else:
            GPIO.output(24, False)
except:
    GPIO.cleanup()
```
Controlling LED with Switch
Controlling LED with Light Sensor

Components:
- LED
- LDR (Light Dependent Resistor)
- Capacitor 1μF
- 2 Resistors (Orange, Orange, Brown, Gold)

```python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
ldr_threshold = 30000
LDR_PIN = 12
LIGHT_PIN = 21

def readLDR(PIN):
    reading = 0
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, False)
    time.sleep(0.1)
    GPIO.setup(PIN, GPIO.IN)
    while (GPIO.input(PIN) == False):
        reading = reading + 1
    return reading

def switchOnLight(PIN):
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, True)

def switchOffLight(PIN):
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, False)

while True:
    try:
        ldr_reading = readLDR(LDR_PIN)
        print(ldr_reading)
        if ldr_reading > ldr_threshold:
            switchOnLight(LIGHT_PIN)
        else:
            switchOffLight(LIGHT_PIN)
        time.sleep(1)
    except KeyboardInterrupt:
        exit()
```
Controlling LED with Light Sensor
Temperature & Humidity Monitoring

• Components:
  • DHT11 or DHT22 Sensor
    • VCC (+)
    • GND (-)
    • DAT (data)

• Install some libraries
  • sudo apt-get update
  • sudo apt-get upgrade
  • sudo apt-get install python3-dev python3-pip
  • sudo python3 -m pip install --upgrade pip setuptools wheel
  • sudo pip3 install Adafruit_DHT
Temperature & Humidity Monitoring

The circuit:
DAT : to GPIO
VCC : to Power
GND : to Ground

```python
import Adafruit_DHT

DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN =

while True:
    try:
        humidity, temperature =
        Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
        if humidity is not None and temperature is not None:
            print("Temp={0:0.1f}*C
Humidity={1:0.1f}%".format(temperature, humidity))
        else:
            print("Failed to retrieve data from humidity
c sensor")
    except KeyboardInterrupt:
        exit()
```
Temperature & Humidity Monitoring
Temperature & Humidity Monitoring
Troubleshoot

• If you find an error like this one:

```bash
temperature.py: File not found
```
Temperature & Humidity Monitoring

Troubleshoot

• After system updates, the hardware name in the /proc/cpuinfo on raspberry pi4 has been changed.

So, it is necessary to edit a platform_detect.py in /usr/local/lib/python3.7/dist-packages/Adafruit_DHT/ directory. All you need it is add next few strings in def pi_version(): function:

```python
if match.group(1) == 'BCM2711':
    return 3
```
Assignment 1 - Specification

• Objectives:
  • Connect and read data from sensors
  • Connect and write data to control actuators

• Upload to E3 before 10/26 (Mon) 23:59PM
  • Assignment 1 – deliverables
    • Report (2-4 pages)
      • Explain the objective
      • Explain your source code and the detail of how your script can read and write your sensors and actuators, respectively
    • Source code
    • 3-minute demo video
  • 1-page project proposal
    • Topic, objective, and sensors/actuators
    • Specs for demos 1, 2, 3, and final demo
  • Zip the above 4 files into one compressed file and upload

• Q&A? post on E3 discussion board
The schedule to pick the Sensors / Actuators

for sensors/actuators and wires

10/7 (Wed) 2-5 PM

MIRC 701 (電資大樓 701)
Thank You