Assignment 1 – Programming Raspberry Pi to Read/Write Sensors and Actuators

Controlling LED with a switch, an LDR sensor, and a DHT sensor
Outline

• Objectives
• Basic Raspberry Pi
• OS installation
• A simple IoT application
  • Controlling LED
    • With Raspberry Pi
    • With Switch
    • With Light Sensor (LDR)
    • With Temperature and Humidity Sensor (DHT)
• Assignment 1 - Specification
Objectives

• Get to know Raspberry Pi
• Capable to install the Raspbian operating system
• Connecting sensors and actuators to Raspberry Pi
• Writing the code to run the sensors and actuators
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 × USB 3.0 ports 2 × USB 2.0 ports.

**GPIO:** Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
Basic Raspberry Pi – GPIO Pins

• GPIO: a physical interface between the Raspberry Pi and the outside world
• To monitor and control the sensors and actuators
• 5 parts of GPIO pins:
  • GPIO: Input/output pins
  • Ground: Zero volts
  • 3.3v: These pins provide 3.3V power
  • 5v: These pins provide 5V power
  • ID EEPROM: Advanced use only
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

• 3 parts of breadboard:
  • The red line or positive: commonly used for Power
  • The blue line or negative: commonly used for Ground
  • The main part in the middle: to put and assemble your sensors or actuators
Basic Raspberry Pi – Sensors

• What are sensors
  • Add almost-human sensing capabilities
  • Take real-world events
  • Convert 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
  • Other Modules, Components and Sensors
Basic Raspberry Pi – Actuators

• What are 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

• Difference in sensors and actuators
  • Sensors: read and get the information from sensors
  • Actuators: write and control some tools based on the previous information

![Diagram showing the process of sensor data moving through a control center to an actuator.]

- Sensor: Temperature sensor detects heat.
- Control Center: Sends detect signal to the control center.
- Actuator: Sprinkler turns on and puts out flame.
Basic Raspberry Pi – Sensors and Actuators

How to get it?

• Borrow from us
  • We have a limited number of 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
    • 抬頭: 國立陽明交通大學
    • 統編: 87557573
Basic Raspberry Pi – Sensors and Actuators
Our Collections of IoT Equipments

<table>
<thead>
<tr>
<th>Item</th>
<th>Total</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raspberry Pi</td>
<td>37</td>
<td>Tiny computer that can be used as the main IoT system</td>
</tr>
<tr>
<td>Power Supply</td>
<td>37</td>
<td>To supply power to Raspberry Pi</td>
</tr>
<tr>
<td>SD Card</td>
<td>37</td>
<td>OS and storage holder</td>
</tr>
<tr>
<td>SD Card Reader</td>
<td>13</td>
<td>Converter from SD Card to micro SD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item</th>
<th>Total</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>BME280</td>
<td>1</td>
<td>Measures the temperature &amp; humidity</td>
</tr>
<tr>
<td>DHT 11</td>
<td>28</td>
<td>Sound Detector</td>
</tr>
<tr>
<td>HW484</td>
<td>1</td>
<td>Gas Leakage Detection</td>
</tr>
<tr>
<td>IR-08H</td>
<td>2</td>
<td>Object Detection</td>
</tr>
<tr>
<td>MH Flying Fish</td>
<td>1</td>
<td>Measure the Distance</td>
</tr>
<tr>
<td>SRD4</td>
<td>1</td>
<td>Small fan to cool small area</td>
</tr>
<tr>
<td>LDR</td>
<td>61</td>
<td>Light Intensity</td>
</tr>
<tr>
<td>PIR</td>
<td>20</td>
<td>Motion Detection</td>
</tr>
<tr>
<td>Water Sensor</td>
<td>1</td>
<td>Detect the presence of water</td>
</tr>
<tr>
<td>Soil Sensor</td>
<td>3</td>
<td>Measure the amount of water in the soil</td>
</tr>
<tr>
<td>Rain Sensor</td>
<td>1</td>
<td>Switching device when rain is detected</td>
</tr>
<tr>
<td>Touch Sensor</td>
<td>1</td>
<td>When touch is closed switch, untouch is open switch</td>
</tr>
<tr>
<td>Raspberry Pi Cam</td>
<td>5</td>
<td>Capturing any activities</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item</th>
<th>Total</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mic</td>
<td>1</td>
<td>Capturing any sound</td>
</tr>
<tr>
<td>RFID Reader</td>
<td>2</td>
<td>A device used to gather information from an RFID tag</td>
</tr>
<tr>
<td>RFID Tags</td>
<td>3</td>
<td>RFID tags are a type of tracking system</td>
</tr>
<tr>
<td>Push ON button</td>
<td>17</td>
<td>Switching device, closed circuit when pushed, open circuit when unpushed</td>
</tr>
<tr>
<td>Switch</td>
<td>7</td>
<td>ON/OFF Switch</td>
</tr>
<tr>
<td>Keypad</td>
<td>2</td>
<td>Record number that is inputted</td>
</tr>
<tr>
<td>Display</td>
<td>8</td>
<td>To display any information given</td>
</tr>
<tr>
<td>Number Display</td>
<td>10</td>
<td>To display number information</td>
</tr>
<tr>
<td>Buzzer</td>
<td>20</td>
<td>To give ring sound</td>
</tr>
<tr>
<td>Fan</td>
<td>9</td>
<td>Small fan to cool small area</td>
</tr>
<tr>
<td>LED</td>
<td>50</td>
<td>To provide lighting</td>
</tr>
<tr>
<td>Bread Board</td>
<td>19</td>
<td>To help developing IoT environment</td>
</tr>
<tr>
<td>Relay</td>
<td>5</td>
<td>Controls the opening and closing of the circuit</td>
</tr>
<tr>
<td>Potentiometer</td>
<td>2</td>
<td>Measures the distance or displacement of an object in a linear</td>
</tr>
<tr>
<td>Capacitor</td>
<td>50</td>
<td>Device that stores electrical energy in an electric field</td>
</tr>
<tr>
<td>Resistor</td>
<td>50</td>
<td>Limit the flow of electric current</td>
</tr>
<tr>
<td>Battery</td>
<td>10</td>
<td>To give power</td>
</tr>
</tbody>
</table>
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 the OS

• Download Raspbian from here:

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

• Here, we choose Raspberry Pi OS (32-bit) with desktop and recommended software since we have 32GB SD card
OS Installation – Tools to Flash the OS (1)

• Right now pi 4b only supports 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 an SD card reader to help you complete this step.)
OS Installation - Tools to Flash the OS (2)
OS Installation – SSH (1)

• 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 (2)

• 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@raspberry.local” default password: raspberry

```bash
C:\Users\chenj>ssh pi@raspberrypi.local
pi@raspberrypi.local's password:
```

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`

Step 1. Choose 5 Interfacing Options -> P3 VNC -> Yes (是)

Step 2. 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”
OS Installation - Notes

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

```
mib3stopierpi@:: $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 169.254.98.144 netmask 255.255.0.0 broadcast 169.254.255.255
      inet6 fe80::1b0b:8f29:bb0b:9fc2 prefixlen 64 scopeid 0x20<link>
      ether dc:a6:32:11:24:39 txqueuelen 1000 (Ethernet)
      RX packets 2722 bytes 303582 (286.4 KiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 3995 bytes 2441752 (2.3 MiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
      loop txqueuelen 1000 (Local Loopback)
      RX packets 17 bytes 1084 (1084.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 17 bytes 1084 (1084.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.0.169 netmask 255.255.255.0 broadcast 192.168.0.255
      inet6 fe80::d833:250f:226c prefixlen 64 scopeid 0x20<link>
      ether dc:a6:32:11:24:15a txqueuelen 1000 (Ethernet)
      RX packets 1451 bytes 255163 (249.1 KiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 110 bytes 11534 (11.2 KiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```
OS Installation - Notes

If you re-install the Raspberry Pi again, maybe you will encounter some problem when you use 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:bb0b:9fc2%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:XgkJTS/4sSSrCkA2A2n2jy/4gW3EhDIwRyso4Yb0TCo.
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 (Y/N)? Y
A Simple IoT Application
Controlling LED with a switch, an LDR sensor, and a DHT sensor
Controlling LED with Raspberry Pi

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)

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

There are two different models 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

Demo video: https://youtu.be/77u4gbw1fVw
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

Demo video:
https://youtu.be/M9R75bi-ahA
Controlling LED with LDR 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 LDR Sensor

Demo video: https://youtu.be/uyhHY2IFaq4
Controlling LED with DHT Sensor

• 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
import Adafruit_DHT
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 21
humidity_threshold = 65
LIGHT_PIN = 20

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:
        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 sensor")
        if humidity > humidity_threshold:
            switchOnLight(LIGHT_PIN)
            print("SWITCH ON LIGHT")
        else:
            switchOffLight(LIGHT_PIN)
        time.sleep(1)
    except KeyboardInterrupt:
        exit()
Controlling LED with DHT Sensor

Demo video:
https://youtu.be/0fJ5V0zgsNo
Controlling LED with DHT Sensor

Troubleshoot

• If you find an error like this one:

```
pi@raspberrypi:~/Documents $ python3 temperature.py
Traceback (most recent call last):
  File "temperature.py", line 8, in <module>
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
File "/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/common.py", line 94,
in read_retry
  humidity, temperature = read(sensor, pin, platform)
File "/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/common.py", line 80,
in read
  platform = get_platform()
File "/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/common.py", line 60,
in get_platform
  from . import Beaglebone_Black
File "/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/Beaglebone_Black.py",
line 24, in <module>
  from . import Beaglebone_Black_Driver as driver
ImportError: cannot import name 'Beaglebone_Black_Driver' from 'Adafruit_DHT' (/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/__init__.py)
```
Controlling LED with DHT Sensor
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 is to add next few strings in def pi_version(): function:

```python
    elif 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 10/27 23:59PM
  • Assignment 1 – deliverables
    • Report (2-4 pages) in PDF and use the template that we provide
      • Explain the objective
      • Explain your source code and the detail of how your script can read and write your sensors and actuators, respectively
      • Link to a 3-minute demo video on YouTube
    • Source code
    • 1-page project proposal in PDF
      • Topic, objective, and sensors/actuators
      • Specs for Assignments 1, 2, and 3 (grow your IoT application instead of changing it for each assignment: Assignments 2 needs to store data to the cloud, while Assignment 3 needs to have two devices and run a program in the cloud)
    • Zip the above 3 files into one compressed file and upload

• Q&A? Post on E3 discussion board
Assignment 1 - Specification

• Note for Assignment 1:
  • You must use different combinations of sensors and actuators than the ones we present
    • If using the same combinations, your maximum score is only 65.
  • The report can be written in Chinese or English, but the video must be delivered in English.
  • In video, explain how you assemble your sensors and actuators, how you connect it, and also explain your source code, and show the results.
  • Upload your video to YouTube and put the link into your report. Don’t upload your video to E3.
  • In your report, make sure you have a diagram of connected sensors and actuators on your board (you can refer the diagram on page 29 and you can use “Fritzing” to draw it).
Enjoy it.