

IFN649 ADVANCED NETWORKS

PRESENTATION

>Unit Coordinator: Professor Raja Jurdak

>Lecturer: Dr. Gowri Ramachandran

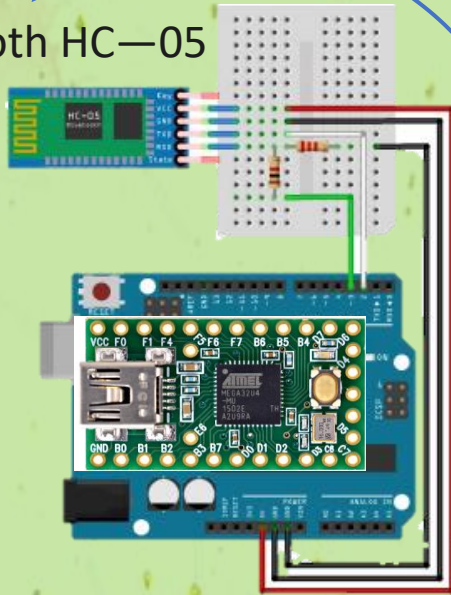
>Tutor: Mr. Tao Liu

>Student Name : Yin Chung HO

>Student ID: 11206322

Sensing tier

Bluetooth HC—05



Teensy On beensy PCB board

Program
in
Teensy

```
sketch_apr26a | Arduino 1.8.12
File Edit Sketch Tools Help
sketch_apr26a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Display "LED_ON"

Upstream communication

Edge tier



Raspberry Pi

Client.py
Subscriber

Publish.py
Publisher

Cloud tier



MQTT on AWS EC2
Cloud

Publish "LED_ON"



Web Client Devices
Publisher

Receives "LED_ON"

Publish "LED_ON"

Downstream communication

T2 SEND TEMP. DATA TO LAPTOP FROM TEENSY

- ✓ The main purpose of this code (C++) is to **collect environmental data** (such as **Temperature** and **Humidity**) from the **DHT11 sensor** through the **Bluetooth module (HC-05)** and **transmit this data to other devices (PC)**.
- ✓ At the same time, **LED** indicators are used to display the **data transmission status**.
- ✓ The Bluetooth **communication rate is set to 9600 bps** and the **LED lights up while transmitting data**.

```
#include "DHT.h"
```

Import DHT library

```
//#include <SoftwareSerial.h>
```

```
#define DHTPIN 21          // Digital pin connected to the DHT sensor  
sensor
```

Set the digital pin connected to the

```
#define DHTTYPE DHT11      // DHT 11
```

sets the sensor type

```
#define LEDPIN 11
```

defines digital pin 11 controls LED, used to indicate if the BT module is sending data

```
DHT dht(DHTPIN, DHTTYPE);
```

Initialize DHT sensor

```
// Teensy 5V <--> HC-05 Vcc
```

```
// Teensy Ground <--> HC-05 GND
```

```
#define rxPin 7 // Teensy pin 7 <--> HC-05 Tx
```

Define the receiving and transmitting pins

```
#define txPin 8 // Teensy pin 8 <--> HC-05 Rx
```

connected to the Bluetooth module HC-05

```
//SoftwareSerial BTSerial = SoftwareSerial(rxPin, txPin);
```



```

void setup() {
    // Setup serial for monitor
    Serial.begin(9600);

    // Setup DHT Sensor
    pinMode(DHTPIN, INPUT);
    dht.begin();

    // Setup Serial1 for Bluetooth
    Serial1.begin(9600);
    // Default communication rate of the Bluetooth module
}

void loop() {
    if(Serial1.available() == 0){
        // Checks whether data is coming from the serial port
        digitalWrite(LEDPIN, HIGH); LED
        float h = dht.readHumidity();
        float t = dht.readTemperature();
        float f = dht.readTemperature(true);

        float hif = dht.computeHeatIndex(f, h);
        float hic = dht.computeHeatIndex(t, h, false);

        Serial.print(F(" Humidity: "));
        Serial.print(h);
        Serial.print(F("% Temperature: "));
        Serial.print(t);
        Serial.print(F("C "));
        Serial.print(f);
        Serial.print(F("F Heat index: "));
        Serial.print(hic);
        Serial.print(F("C "));
        Serial.print(hif);
        Serial.println(F("F"));

        Serial1.print(F(" Humidity: "));
        Serial1.print(h);
        Serial1.print(F("% Temperature: "));
        Serial1.print(t);
        Serial1.print(F("C "));
        Serial1.print(f);
        Serial1.print(F("F Heat index: "));
        Serial1.print(hic);
        Serial1.print(F("C "));
        Serial1.print(hif);
        Serial1.println(F("F"));

        delay(1000);
        digitalWrite(LEDPIN, LOW);
        delay(1000);
    }
}

```

View data via the serial monitor of the Arduino IDE

comm. rate 9600 bps

Output data to monitor

Set the sensor pin to input
and enable the DHT sensor

Transfer data via BT

Turn on to indicate data transfer

Read sensor data

Delay and turn off LED

T3 SEND TEMP. DATA TO RASPBERRY PI FROM TEENSY

The code is used to **continuously monitor** the **Bluetooth serial port** (/dev/rfcomm0) and **communicate with the Arduino**.

When communication starts, the code sends the "Start\r\n" command to the Arduino.

Afterwards, it **continuously checks** to see **if there is data** to read from the serial port, **reads the data**, then **decodes and displays it on the console**.

```
import serial
```

匯入必要的庫

```
import time
```

```
import string
```

```
# reading and writing data from and to arduino serially.
```

```
# rfcomm0 -> this could be different
```

```
ser = serial.Serial("/dev/rfcomm0", 9600)
```

設定串口連接,通信速率為 9600 bps

```
ser.write(str.encode('Start\r\n'))
```

向 Arduino 發送初始訊息

```
while True:
```

主循環

```
    if ser.in_waiting > 0:
```

檢查是否有數據可讀取

```
        rawserial = ser.readline()
```

從串口讀取一行數據

```
        cookedserial = rawserial.decode('utf-8').strip('\r\n')
```

將讀取的字節數據解碼為 UTF-8 格式的字符串

```
        print(cookedserial)
```

將處理後的字符串輸出到控制台

TS CLIENT → SUBSCRIBER

- ✓ This code subscribes to a specified **MQTT topic “ifn649”** and **sends the received messages** to a hardware device (**Arduino**).

```
import paho.mqtt.client as mqtt
```

Import client libraries handle the
MQTT protocol

```
import serial
```

```
def on_connect(client, userdata, flags, rc):
```

```
# Function for making connection
```

```
print("Connected to MQTT")
```

```
print("Connection returned result: " + str(rc) )
```

```
client.subscribe("ifn649")
```

Called when the client
successfully connects

Subscribe to topic "ifn649 "

```
def on_message(client, userdata, msg):
```

```
# Function for Sending msg
```

```
print(msg.topic+" "+str(msg.payload))
```

```
ser = serial.Serial("/dev/rfcomm0", 9600)
```

```
ser.write(str.encode(str(msg.payload)))
```

When rec. a msg from a sub. topic

Print msg subject and payload

```
client = mqtt.Client()
```

```
client.on_connect = on_connect
```

```
client.on_message = on_message
```

```
client.connect("test.mosquitto.org", 1883, 60)
```

```
client.loop_forever()
```

Create an MQTT client

Connect MQTT client to proxy server

Start MQTT network loop

TS PUBLISHER → PUBLISHER

- ✓ This code uses the **publish** module from the MQTT library to publish an MQTT message "**Hello World**" to the specified **topic "ifn649"** Subscriber to the **MQTT broker server** at **test.mosquitto.org** by using the MQTT protocol.

```
import paho.mqtt.publish as publish
```

Import the publish module

```
publish.single("ifn649", "Hello World", hostname="test.mosquitto.org", port=1883)
```

"ifn649"=MQTT topic, "Hello World"= Msg to be published , hostname = Proxy server

```
print("Done")
```

TS CONTROLLER (ARDUINO)

- ✓ It controls the switch of an LED and receives instructions via Bluetooth connect with Raspberry Pi to operate the LED.

```
#define LEDPIN 11

void setup() {
    // Setup serial for monitor and Setup Serial1 for BlueTooth
    Serial.begin(9600);
    Serial1.begin(9600);
    pinMode(LEDPIN, OUTPUT);
}


```

Initialize serial port

for monitoring

Bluetooth module

Config LED pins for output mode

```
void loop() {
    // Process commands from bluetooth first.
    if(Serial1.available() > 0){
        String str = Serial1.readString().substring(2);
        str = str.substring(0, str.length() - 1);
        //Serial.println(str);
        if(str == "LED_ON"){
            digitalWrite(LEDPIN, HIGH);
            Serial.println("LED ON");
        }
        else if(str == "LED_OFF"){
            digitalWrite(LEDPIN, LOW);
            Serial.println("LED is OFF");
        }
    }
}


```

Receive commands from BT

Check if there is data readable from the BT serial port

If the rec. str. is "LED_ON" light up LED

Print "LED ON" on the monitor