

物聯網導論期末專題

6 小孩 0 負擔

第 8 組

R08525116 吳承哲

R08942088 李政旻

一、動機與目的

生活中，父母帶小孩出門逛街或是出遊時，總是有可能發生小孩走失的意外，更有甚者，這個意外造成無法彌補的遺憾；又或是帶著寵物出外散步郊遊時，有些人會帶到野外，解開繩索讓他奔跑，如果一時不察就需要花很多心力找回，也可能直接就被有心人士帶走。

有鑑於此，我們想做出一個穿戴式的定位裝置，能夠讓孩童佩戴在身上，父母能夠上網配對該裝置的 ID，在外出時給小孩帶著，為了在走失時能夠及時找到小孩的位置，或者能夠在寵物走失時，即時追蹤自己寵物的位置。

因此，我們將客群鎖定在各位爸媽，而小孩年齡設定在會走路到會用手機之前的時期。另外，也能將此商品穿戴在寵物身上。

二、 分析與評估

2019 年 2 月初，歐盟宣佈全面回收德國品牌 ENOX 推出的 Safe-KID-One 兒童智慧手錶。《BBC》報導，回收主因是這款手錶「有嚴重隱私漏洞」。

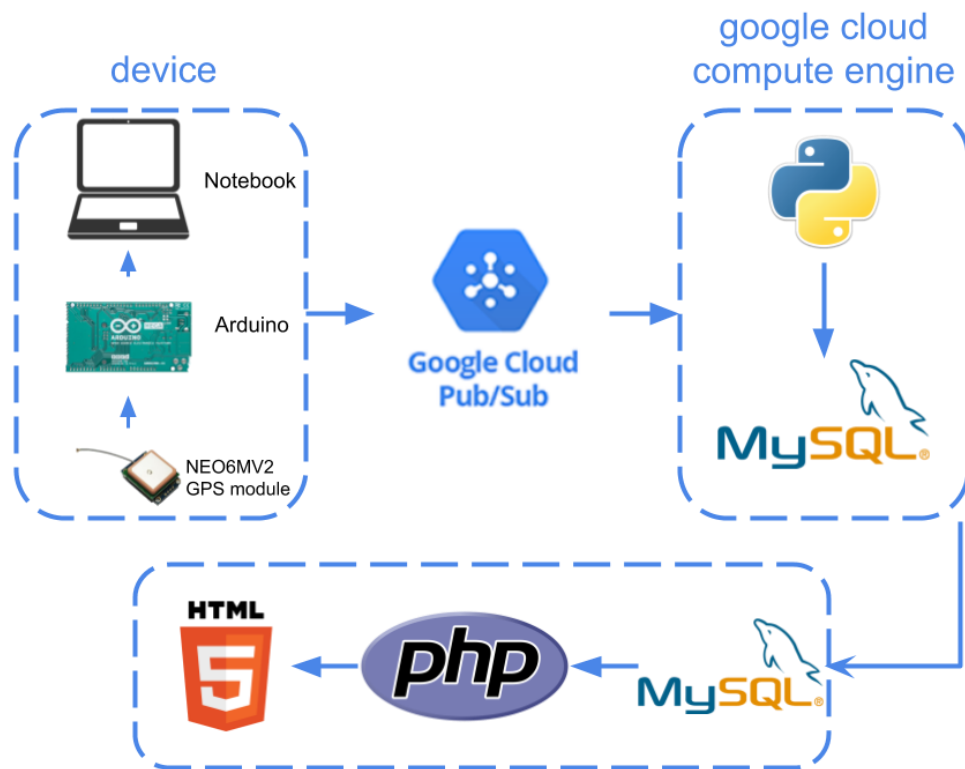
歐盟快速警告系統 RAPEX 指出，這款手錶在傳送訊息或儲存資料時沒有加密，讓不法份子能任意取得伺服器數據、追蹤配戴兒童位置，甚至任意擷取、修改定位，讓手錶發送錯誤定位給家長，「嚴重影響兒童人身安全。」

其實早在 2017 年 10 月，挪威消費者理事會（NCC）就聘用一家安全公司測試歐洲市面上販售的多款兒童智慧手錶，結果只有「Tinitel」這款沒有安全漏洞，其他都可以輕易被駭客入侵。當時 NCC 發言人尼爾（Alex Neill）提醒家長，謹慎評估使用這類手錶的必要性，「安全應該是第一要務。如果連這一點都無法保障，根本不該賣這些產品。」

於是我們便將目標放在加密傳輸上，由於 Google 不僅提供 Google map 的使用介面，也另外有 IoT 相關的輔助，最為重要的是使用他們的 API 都必須經過加密才能傳輸，完全符合我們

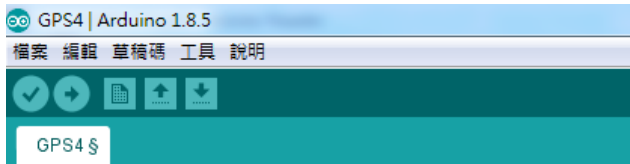
的需求，所以本專案就以 GPS 模組和 Google 雲端服務做為模擬。

三、 材料與系統架構



四、 實作

首先，透過 GPS 裝置取得位置資訊，並藉由 arduino 將資訊取出，包含經緯度座標以及時間戳，但由於取得的時間戳時區不同且較為不準確，故改而使用 python 取得時間。



```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

// Choose two Arduino pins to use for software serial
int RXPin = 2;
int TXPin = 3;

int GPSBaud = 9600;

// Create a TinyGPS++ object
TinyGPSPlus gps;

// Create a software serial port called "gpsSerial"
SoftwareSerial gpsSerial(RXPin, TXPin);

void setup()
{
  // Start the Arduino hardware serial port at 9600 baud
  Serial.begin(9600);

  // Start the software serial port at the GPS's default baud
  gpsSerial.begin(GPSBaud);
}
```

```
void displayInfo()
{
  if(gps.location.isValid() && gps.date.isValid() && gps.time.isValid())
  {
    Serial.print(gps.location.lat(), 6);
    Serial.print(',');
    Serial.print(gps.location.lng(), 6);
    Serial.print(',');
    Serial.print(gps.date.year());
    Serial.print("-");
    Serial.print(gps.date.month());
    Serial.print("-");
    Serial.print(gps.date.day());
    Serial.print(" ");

    if (gps.time.hour() < 10) Serial.print(F("0"));
    Serial.print(gps.time.hour());
    Serial.print(":");
    if (gps.time.minute() < 10) Serial.print(F("0"));
    Serial.print(gps.time.minute());
    Serial.print(":");
    if (gps.time.second() < 10) Serial.print(F("0"));
    Serial.println(gps.time.second());
  }
  else
  {
    Serial.println("Not Available");
  }
}
```

再藉由 Google pub/sub 的服務發布到 Google 雲端上。

```
if __name__ == "__main__":
    parser = argparse.ArgumentParser(
        description=__doc__,
        formatter_class=argparse.RawDescriptionHelpFormatter,
    )
    parser.add_argument("project_id", help="Google Cloud project ID")
    parser.add_argument("topic_name", help="Pub/Sub topic name")

    args = parser.parse_args()

    plist = list(serial.tools.list_ports.comports())
    if len(plist) <= 0:
        print("The Serial port can't find!")
        exit()
    else:
        plist_0 = list(plist[0])
        serialName = plist_0[0]
        ser = Serial(serialName, 9600, timeout=.5)

    while True:
        if ser.inWaiting():
            str = ser.readline().decode('utf8').split(",")
            if len(str) == 3:
                str[2]=str[2][:-2]
                print(str)
                payload = json.dumps({'name': 'test', 'lat': str[0], 'lng': str[1],
                    'time': time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())}).encode(encoding="utf-8")

                pub(args.project_id, args.topic_name, payload)
            else:
                print("not available.")
        time.sleep(1)
```

接著在 Google 開啟的 VM 中架設好 MySQL 的服務，並訂閱剛

才發布的那個主題，將其中的經緯度資訊存入資料庫中。

```
def sub(project_id, subscription_name):
    """Receives messages from a Pub/Sub subscription."""
    # [START pubsub_quickstart_sub_client]
    # Initialize a Subscriber client
    client = pubsub_v1.SubscriberClient()
    # [END pubsub_quickstart_sub_client]
    # Create a fully qualified identifier in the form of
    # `projects/{project_id}/subscriptions/{subscription_name}`
    subscription_path = client.subscription_path(project_id, subscription_name)

def callback(message):
    print(
        "Received message {} of message ID {}".format(
            message, message.message_id
        )
    )
    # Acknowledge the message. Unack'd messages will be redelivered.
    message.ack()
    print("Acknowledged message {}".format(message.message_id))

    global db
    cursor = db.cursor()
    payload = json.loads(message.data)
    #print("insert into position values('{0}',{1},{2},{3})".format(payload["name"], payload["lat"], payload["lng"], payload["time"]))
    try:
        sql = "insert into position values('{0}',{1},{2},{3})".format(payload["name"], payload["lat"], payload["lng"], payload["time"])
        #print("sql: "+sql)
        cursor.execute(sql)
        db.commit()
    except:
        print("insert failed")
        db.rollback()

streaming_pull_future = client.subscribe(
    subscription_path, callback=callback
)
print("Listening for messages on {}.\\n".format(subscription_path))

# Calling result() on StreamingPullFuture keeps the main thread from
# exiting while messages get processed in the callbacks.
try:
    streaming_pull_future.result()
except: # noqa
    streaming_pull_future.cancel()

if __name__ == "__main__":
    parser = argparse.ArgumentParser(
        description=__doc__,
        formatter_class=argparse.RawDescriptionHelpFormatter,
    )
    parser.add_argument("project_id", help="Google Cloud project ID")
    parser.add_argument("subscription_name", help="Pub/Sub subscription name")

    args = parser.parse_args()

    try:
        db = pymysql.connect("localhost","IoT","IoT","IoT")
    except:
        print("Connect failed")
        exit()

    sub(args.project_id, args.subscription_name)

    while True:
        time.sleep(60)
# [END pubsub_quickstart_sub_all]
```

然後需要一個取用 MySQL 資料的介面，這部分一樣在 Google

的 VM 中，建立一個 PHP 頁面接受指令，以供之後網頁取得。

```

// select, data = name
else if($_REQUEST["state"] == "select"){
    // convert type to array
    if(gettype($_REQUEST["data"]) == "string"){
        $data = $_REQUEST["data"];
    }
    else{
        $state["error"] = "data";
        $state["message"] = "Unsupport data type.";
        $state["received"] = $_REQUEST["data"];
        die(json_encode($state));
    }

    try{
        // prepare sql
        $PDOconn = DBconnect_PDO();
        // set the PDO error mode to exception
        $PDOconn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        // prepare sql and bind parameters
        $PDOstmt = $PDOconn->prepare("SELECT * FROM `position` WHERE name=:name ORDER BY timestamp DESC LIMIT 1");
        $PDOstmt->bindParam(':name', $data);

        // insert a row
        $PDOstmt->execute();

        // fetch row
        $row = $PDOstmt->fetch(PDO::FETCH_ASSOC);
    }
    catch(PDOException $e){
        $state["error"] = "mysql";
        $state["message"] = $e->getMessage();
        die(json_encode($state));
    }
    catch(Exception $e){
        $state["error"] = "mysql";
        $state["message"] = $e->getMessage();
        die(json_encode($state));
    }
    $PDOconn = null;

    echo(json_encode($row));
}
else{
    $state["error"] = "parameters";
    $state["message"] = "Lack of fields.";
    $state["received"] = array_keys($_REQUEST);
    die(json_encode($state));
}
}

```

最終，透過一個網頁使用 Google map API 在地圖上顯示裝置的位置。

```

var last = {"timestamp":0};
var marker;

function initMap() {
    var map = new google.maps.Map(document.getElementById('map'), {
        center: new google.maps.LatLng(25.02014, 121.553582),
        zoom: 12
    });
    marker = new google.maps.Marker({map:map});

    setInterval(function(){
        $.ajax({url:"mysqlInterface.php?state=select&data=test", type:"post",
            success:function(data, textStatus){
                var position = JSON.parse(data);
                //console.log(position);

                // update point when position have been changed
                if(position["timestamp"] != last["timestamp"]){
                    var point = new google.maps.LatLng(
                        parseFloat(position["lat"]),
                        parseFloat(position["lng"])
                    );

                    marker.setPosition(point);
                    marker.setIcon({url: "img/mark.gif"});
                    last = position;
                }
            },
            error:function(XMLHttpRequest, textStatus, errorThrown){
                console.log(textStatus + errorThrown);
                return false;
            }
        });
    }, 5000);
}

```

五、 結果與展示

Python 取得 arduino 傳來的經緯度座標資訊。

```

121.5424246, 25.0173750
121.5425159, 25.0174094
121.5426072, 25.0174438

```

將經緯度座標發佈到 Google Pub/Sub 服務上。

```
PS C:\Users\AbCd\Dropbox\一上\物聯網\project> python pub.py spring-archive-254114 GPS
Published message b'{"name": "test", "lat": 25.0190077, "lng": 121.5425054, "time": "2020-01-08 11:53:27"}' now has message ID 922498494382154
Published message b'{"name": "test", "lat": 25.01901149, "lng": 121.54240395, "time": "2020-01-08 11:53:32"}' now has message ID 922498913287084
Published message b'{"name": "test", "lat": 25.01901528, "lng": 121.54230249999999, "time": "2020-01-08 11:53:37"}' now has message ID 922498380446600
Published message b'{"name": "test", "lat": 25.019019070000002, "lng": 121.542201049999999, "time": "2020-01-08 11:53:42"}' now has message ID 922499053542219
Published message b'{"name": "test", "lat": 25.019022860000003, "lng": 121.54209959999999, "time": "2020-01-08 11:53:47"}' now has message ID 922499034316766
Published message b'{"name": "test", "lat": 25.019026650000004, "lng": 121.54199814999998, "time": "2020-01-08 11:53:52"}' now has message ID 922498916410252
Published message b'{"name": "test", "lat": 25.019030440000005, "lng": 121.54189669999998, "time": "2020-01-08 11:53:57"}' now has message ID 922508101150750
Published message b'{"name": "test", "lat": 25.019034230000006, "lng": 121.54179524999998, "time": "2020-01-08 11:54:02"}' now has message ID 922490587934558
Published message b'{"name": "test", "lat": 25.019038020000007, "lng": 121.54169379999998, "time": "2020-01-08 11:54:07"}' now has message ID 922499192809080
Published message b'{"name": "test", "lat": 25.019041810000008, "lng": 121.54159234999997, "time": "2020-01-08 11:54:12"}' now has message ID 922498356566219
Published message b'{"name": "test", "lat": 25.019045600000001, "lng": 121.54149089999997, "time": "2020-01-08 11:54:17"}' now has message ID 922499110937801
Published message b'{"name": "test", "lat": 25.019049390000001, "lng": 121.54138944999997, "time": "2020-01-08 11:54:22"}' now has message ID 922499128199019
Published message b'{"name": "test", "lat": 25.019053180000001, "lng": 121.54128799999997, "time": "2020-01-08 11:54:27"}' now has message ID 922498477312937
```

Google Pub/Sub 服務顯示推播的封包內容。

訊息

如要查看發布至這個主題的訊息，請選擇或建立 (用缺省的建議選項) 提取訂閱項目。

提取 Cloud Pub/Sub 訂閱項目 *

projects/spring-archive-254114/subscriptions/GPS



按一下 [提取] 即可查看訊息並暫時延遲傳遞訊息給其他訂閱者。

如要永久阻止訊息傳遞給其他訂閱者，請提取 [啟用訊息確認] 並點選訊息旁邊的 [確認]。系統一次只會提取少數訊息。只要再次點選 [提取]，即可從待處理工作中提取更多訊息。在實際工作環境中，請謹慎使用這個選項。在您超過確認時限 (10 秒) 後，如果此訂閱的其他訂閱者未確認訊息，系統會再次傳送該訊息。 [瞭解詳情](#)

提取

☐ 啟用訊息確認

篩選表格

發布時間	屬性鍵	訊息內文	內文 JSON 金鑰	body.lat	body.lng	確認 ↑
2020年1月8日 下午2:40:59	—	{'name': 'test', 'lat': 25.019038020000007, 'lng': 121.54169379999998, 'time': '2020-01-08 14:40:59'}	<div>name lat</div> <div>lng time</div>	25.019038020000007	121.5416	已超過確認期限

存入資料庫中的結果。

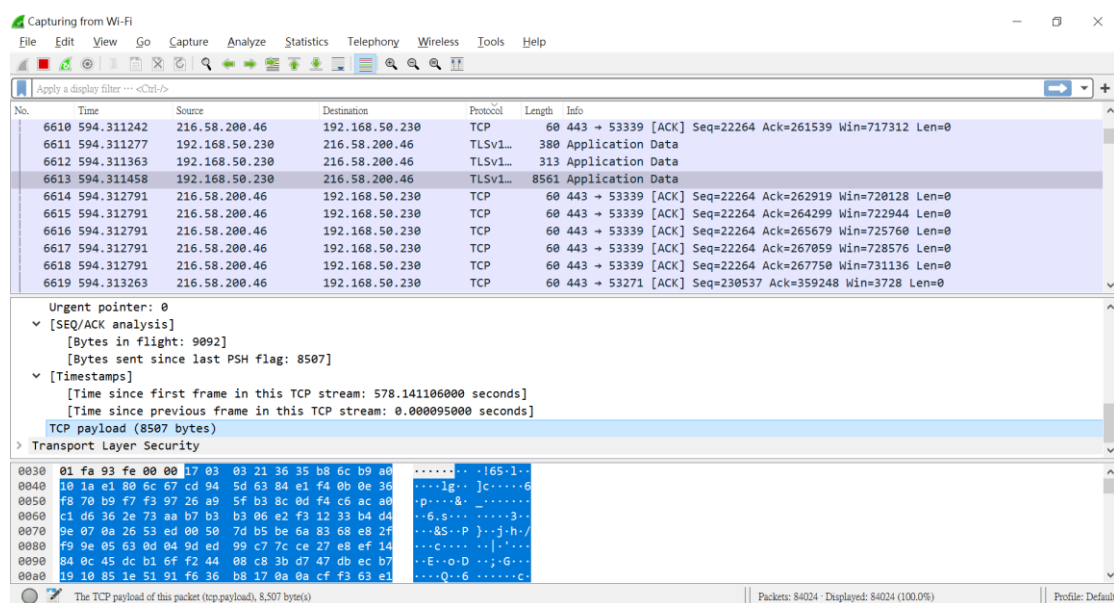
+ 選項

name	lat	lng	timestamp
test	25.019037	121.541695	2020-01-08 14:40:59

在 Google 地圖上以紅點閃爍顯示座標。



接著使用 WireShark 攔截封包，確認傳輸的資料與協定，可以看出通過 TLS 加密傳輸，資料內容是無法直接明文檢視的。



六、 結論與展望

定位手環已經是上市的產品，而最大的問題是在隱私部分，如果資訊遭到攔截，意味著他人可以查詢到你的位置。因此希望利用使用者帳號連結裝置 ID，並經過嚴謹的資訊加密，防止有心人士竊取資訊。

另外，也可以加入無線電通話功能，當小孩走丟時可以即時通話，在走丟當下透過手環聯絡小孩原地等待等等，能夠安撫走丟小孩的情緒，較不會發生再亂跑的情況，也能夠在意外發生時根據現場環境收音及時了解情形。

最後，考慮到家庭經濟跟必要性的因素，此產品可以跟政府生育計畫結合，一個小孩領一隻手環，給新手爸媽帶小孩一個得力助手，在幼童懂事以前，更輕鬆安全的照顧幼童。

七、 參考資料

「我小孩被綁架？」：

<https://news.tvbs.com.tw/local/674197>

大埔尋狗：<https://www.hk01.com/突發/406199/大埔尋狗-細膽狗>

[女寒風中走失-兩日後自行返回住所](#)

有定位功能、可發求救訊號的「兒童智慧型手錶」，為何歐盟要求緊急收回？：

<https://www.storm.mg/lifestyle/1053066>

小米手錶：<https://shopee.tw/小米米兔-兒童手錶 3C-男孩女孩 4g 智能防水定位小愛同學-多功能手環-7 天超長待機-高清視頻通話-i.134070963.2796290900>

Google Map API:

<https://developers.google.com/maps/documentation/java-script/geolocation>

<https://developers.google.com/maps/documentation/java-script/mysql-to-maps>

<https://w3c.github.io/geolocation-api/>

Google pub/sub API:

<https://github.com/GoogleCloudPlatform/python-docs-samples/tree/master/pubsub/cloud-client>

Arduino with GPS Module:

<https://randomnerdtutorials.com/guide-to-neo-6m-gps-module-with-arduino/>

<https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/>