



無線通訊網路 FINAL PROJECT

希望曙光

組內成員：

葉軒 韓觀 張予軒 劉晉瑋 王郁翔

目錄

- 構思背景
- 程式演算法
- 實作應用
- 未來展望

『希望』

- 『希望』是我們的開發版，搭載著接收光敏電阻所製作而成

『曙光』

- 『曙光』是我們設計之演算法之延伸物，也就是此次實測的應用

構思理念

請聽我們娓娓道來

- 在無盡的黑暗中，我們匍匐向前，彷彿在找尋著甚麼.....，徬徨無助，這就是心靈寫照，結束了，一切都結束了.....旅行終結
- 回程.....，我們決定要做出點甚麼，不可以再重蹈覆轍，不可以再次失敗，若有下次，必定者歸來

程式演算法

```
int main()
{
    // object to control the debug board

    plan = new lora::ChannelPlan_AS923();

    dot = mDot::getInstance(plan);

    // set network
    config(dot,plan);

    while (true) {
        int tmp,ret;
        std::vector<uint8_t> data;
        stringstream ss;
        std::string data_str ;

        tmp = in.read_u16();
        printf("%d\r\n",tmp);

        // format data for sending to the gateway
        ss << tmp;
        if(tmp){
            ss << "Day!";
        }
        else{
            ss << "Night!";
        }
        ss >> data_str;
        for (std::string::iterator it = data_str.begin(); it != data_str.end(); it++)
            data.push_back((uint8_t) *it);

        // send the data to the gateway
    }
```

```
tmp = in.read_u16();
printf("%d\r\n",tmp);

// format data for sending to the gateway
ss << tmp;
if(tmp){
    ss << "Day!";
}
else{
    ss << "Night!";
}
ss >> data_str;
for (std::string::iterator it = data_str.begin(); it != data_str.end(); it++)
    data.push_back((uint8_t) *it);

// send the data to the gateway
if ((ret = dot->send(data)) != mDot::MDOT_OK) {
    logError("failed to send\r\n", ret, mDot::getReturnCodeString(ret).c_str());
} else {
    logInfo("successfully sent data to gateway\r\n");
}

// we use US but in the 868 (EU) frequency band, we need to wait until another channel is available before transmitting again
osDelay(std::max((uint32_t)5000, (uint32_t)dot->getNextTxMs()));
}

return 0;
```

```

#if ACTIVE_EXAMPLE == OTA_EXAMPLE
void config(mDot *dot, lora::ChannelPlan *plan)
{
    uint8_t ret;
    // reset to default config so we know what state we're in
    dot->resetConfig();
    // set how many log info will be show
    dot->setLogLevel(mts::MTSLog::INFO_LEVEL);
    // set subband frequency the same as gateway so gateway can listen to you
    logInfo("setting frequency sub band\r\n");
    if ((ret = dot->setFrequencySubBand(config_frequency_sub_band)) != mDot::MDOT_OK) {
        logError("failed to set frequency sub band %d:%s\r\n", ret, mDot::getReturnCodeString(ret).c_str());
    }
    // lora has private network and public network here we use public network
    logInfo("setting public network mode");
    if ((ret = dot->setPublicNetwork(true)) != mDot::MDOT_OK) {
        logError("failed to public network mode");
    }
    std::vector<uint8_t> temp;

    for (int i = 0; i < 8; i++) {
        temp.push_back(config_app_eui[i]);
    }
    // set network id
    logInfo("setting app eui\r\n");
    if ((ret = dot->setNetworkId(temp)) != mDot::MDOT_OK) {
        logError("failed to set app eui %d:%s\r\n", ret, mDot::getReturnCodeString(ret).c_str());
    }
    temp.clear();
    for (int i = 0; i < 16; i++) {
        temp.push_back(config_app_key[i]);
    }
    // set network key

```



```
// set network key
logInfo("setting app key\r\n");
if ((ret = dot->setNetworkKey(temp)) != mDot::MDOT_OK) {
    logError("failed to set app key %d:%s\r\n", ret, mDot::getReturnCodeString(ret).c_str());
}
// a higher spreading factor allows for longer range but lower throughput
// in the 915 (US) frequency band, spreading factors 7 - 10 are available
logInfo("setting TX spreading factor\r\n");
/*if ((ret = dot->setTxDataRate(mDot::SF_10)) != mDot::MDOT_OK) {
    logError("failed to set TX datarate %d:%s\r\n", ret, mDot::getReturnCodeString(ret).c_str());
}*/
// request receive confirmation of packets from the gateway
logInfo("enabling ACKs\r\n");
if ((ret = dot->setAck(1)) != mDot::MDOT_OK) {
    logError("failed to enable ACKs %d:%s\r\n", ret, mDot::getReturnCodeString(ret).c_str());
}
// Set Tx Power
logInfo("enabling Tx Power\r\n");
if ((ret = dot->setTxPower(20)) != mDot::MDOT_OK) {
    logError("failed to enable Tx Power %d:%s\r\n", ret, mDot::getReturnCodeString(ret).c_str());
}
// request receive confirmation of packets from the gateway
logInfo("enabling Tx Data Rate\r\n");
if ((ret = dot->setTxDataRate(3)) != mDot::MDOT_OK) {
    logError("failed to enable Tx Data Rate %d:%s\r\n", ret, mDot::getReturnCodeString(ret).c_str());
}
// save this configuration to the mDot's NVM
logInfo("saving config\r\n");
if (! dot->saveConfig()) {
    logError("failed to save configuration\r\n");
}
logInfo("joining network\r\n");
while ((ret = dot->joinNetwork()) != mDot::MDOT_OK) {
    logError("failed to join network %d:%s\r\n", ret, mDot::getReturnCodeString(ret).c_str());
    osDelay(std::max((uint32_t)1000, (uint32_t)dot->getNextTxMs()));
}
}
```

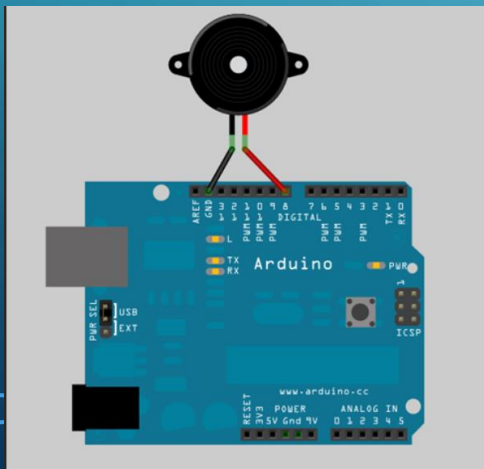

實作應用

- 銜接著先前未完結的故事，我們給出了解答，在某次旅行中，我們進入了山林古蹟，拿著手機，興高采烈的拍攝絕美風景，天色逐漸灰暗，但我們卻被陷入迷人景色的泥淖.....
- 天色昏暗，但我們仍然身處密林！此時，我們所做的『希望曙光』，就可以派上用場
- 藉由『曙光』的幫忙，隨時傳遞光敏電阻訊號至機台，借由指令『Day』、『Night』，來告知出門遠行者別因美景而忘記時間，這便是我們實際的應用

未來展望

- 後續版本更新事項

1. 我們會更新上蜂鳴器，以及定時器，使其在固定時間間格收發訊號，並於收到訊號時發出提示聲音來提醒使用者，避免使用者忘記查看訊息



2. 有鑑於大家眼睛總是介於手機螢幕與景色之間，我們會將收發訊息的開發版與手機程式做連接，連接的方式是使用藍芽裝置，讓使用者在欣賞、紀錄美景之餘，可以隨時收到『希望曙光』，及時回傳的資訊
3. 增加上等級區分，通知使用者目前白天照明程度，等級界在1~8，等級8最亮，等級1最為昏暗，例如:白天 等級7 ---大概等於能見度極佳的程度，讓使用者可以在自覺能見度不高時及早進行撤離的動作，提升安全保障

