

Fonctionnement interne de la fonction begin du shield

LoRa Fabian

1 Juillet 2015

1 Description

La fonction

```
void LoraShield::begin(String dns);
```

sert à modifier le hostname du shield. Son utilisation se fait après un appel à la fonction init() du shield. Par exemple :

```
#include <LoraShield.h>
#include <SPI.h>
```

```
LoraShield lora;
```

```
void setup()
{
  Serial.begin(9600);
  while (!Serial); // wait for serial port to connect. Needed for Leonardo only
  lora.init();
  lora.begin("temperature.s.ackl.io");
}
```

```
void loop()
{ }
```

2 Côté Arduino

La fonction begin est contenu dans *LoraShield.cpp*, voici sa définition :

```
void LoraShield::begin(String dns)
{
  digitalWrite(SS_PIN, LOW);

  // The byte is the status of the last command
  int previous_cmd_status = SPI.transfer(ARDUINO_CMD_DNS);
  delayMicroseconds(WAIT_TIME_BETWEEN_BYTES_SPI);
```

```

// Data size to be sent
// MSB
int shield_status = SPI.transfer(dns.length() >> 8);
delayMicroseconds(WAIT_TIME_BETWEEN_BYTES_SPI);

shield_status = SPI.transfer(dns.length());
delayMicroseconds(WAIT_TIME_BETWEEN_BYTES_SPI);

//Send: payload as bytes to send
for (int i = 0; i < dns.length() ; i++)
{
    // written_data+1 as to avoid the first byte, which is the ARDUINO_CMD_WRITE
    shield_status = SPI.transfer(dns[i]);
    delayMicroseconds(WAIT_TIME_BETWEEN_BYTES_SPI);
}

digitalWrite(SS_PIN,HIGH);
delayMicroseconds(WAIT_TIME_BETWEEN_BYTES_SPI);
}

```

Ainsi le shield va recevoir un message de la forme : |0x08|longueur|hostname|

3 Côté Contiki

3.1 Changement du hostname

La récupération du nouveau hostname se réalise dans le fichier `/platform/lorafabian/apps/arduino_interface/arduino_cmd.c`. Le programme attend un message de type DNS (0x08) et se charge d'écrire le nouvel hostname suivi d'un espace (pour marquer sa fin) dans le fichier `/HOSTNAME_LORA`.

3.2 Récupérer le hostname

Pour récupérer le hostname du moment, il suffit de lire le fichier `/HOSTNAME_LORA` en faisant attention au dernier espace. Voici le contenu de la fonction `updateHOSTNAME()` présente dans `/examples/lorafabian/lorafab_beacon_answer_new/lorafab_beacon_answer_new.c` :

```

#include "cfs/cfs.h"

//The response to the beacon
char coap_payload_beacon[1000];

/**
 * \brief: update the hostname with /HOSTNAME file
 */
void updateHOSTNAME()
{
    char dns[500]; //The content of the file
    int fd;
    //Read in /HOSTNAME_LORA

```

```

fd = cfs_open("/HOSTNAME_LORA", CFS_READ);
if(fd >= 0) {
    //Read 500 char
    cfs_read(fd, dns, sizeof(dns));
    cfs_close(fd);
    //Get the real hostname
    int size = 1;
    //Because the space significate the end of the hostname
    while(dns[size] > ' ')
        ++size;
    //final = the real url
    char final[size];
    int i;
    for(i = 0; i != size; ++i)
        final[i] = dns[i];
    strcpy(coap_payload_beacon, "{\n\n:}");
    strcat(coap_payload_beacon, final);
    strcat(coap_payload_beacon, "\n}");
}
else {
    printf("ERREUR LORS DE LA LECTURE\n\nr");
    strcpy(coap_payload_beacon, "{\n\n:default.test}");
    return;
}
printf("HOSTNAME : %s\n\nr", coap_payload_beacon);
}

```

Note : Si le fichier `/HOSTNAME_LORA` n'est pas encore écrit, le hostname sera `default.test`.